MORTY: A Toolbox for Mode Recognition and Tonic Identification

Altuğ Karakurt
Bilkent University
1932 Wallamaloo Lane
Wallamaloo, New Zealand
trovato@corporation.com

Sertan Şentürk,
Universitat Pompeu Fabra
Roc Boronat, 138
Barcelona, Spain 08018
sertan.senturk@upf.edu

Xavier Serra
Universitat Pompeu Fabra
Roc Boronat, 138
Barcelona, Spain 08018
xavier.serra@upf.edu

ABSTRACT

CCS Concepts

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

Keywords

Mode recognition; Tonic Identification; Toolbox; Ottoman-Turkish makam music; Carnatic Music; Hindustani Music; Pitch Class Distribution; k-nearest neighbors

1. INTRODUCTION

In many music cultures, the mode specifies the scale and melodic characteristics of the music. Mode recognition is an important task for computational musicology to study the underlying phenomena behind this concept. Tonic is the reference frequency of a recording, which is used to determine the frequencies that correspond to the notes of the studied musical tradition. Many modal music cultures don't have a consensus on frequency values for notes, which makes it crucial to identify the tonic frequency correctly. Knowing it allows us to reveal the mapping of notes to frequencies and to computationally study pitch-related properties of a recording. This mapping also makes it possible to compare such properties of multiple recordings independent from their potentially different tonic frequencies. Hence, estimating the tonic of a recording is the first step for various tasks, such as tuning analysis [8], and automatic transcription [5]. One of our motivations for this work is to provide dependable tools to be used for such higher level tasks for modal music cultures.

We present MORTY, an open source toolbox for mode recognition and tonic identification. It contains three pitch histogram based methods, two of which are state of the art algorithms proposed for specific modal music traditions. We provide generalized implementations of these that can be used for any modal music culture, as well as our multi-layer perceptron neural network specialized for mode recognition. Our neural network implementation uses the pitch histogram of recordings to recognize their modes.

We consider unavailability of public tools and datasets, as well as lack of reproduciblity to be some of the biggest obstacles in front of the progress of not only computational musicology, but for computational research in general. Our aim is to propose our primary work on applying neural networks on modal music information retrieval and to pave the way for future research on modal music, by providing the open source implementations of state of the art algorithms and our approach together with a public dataset for evaluation. Accessibility of such tools and data would help the researchers quantify the performances of their novel algorithms and compare them with the state-of-the-art.

Our contributions can be summarized as:

1. An open toolbox aimed to set a benchmark for future research in mode recognition and tonic identification, which implements and generalizes the state of the art methodologies proposed by Bozkurt and Gedik [7, 13], and Chordia and Şentürk [10]
2. The largest makam recognition dataset for Ottoman-Turkish Makam Music (OTMM), composed of 1000 audio recordings from 20 makams with annotated tonic frequency and editorial metadata.
3. Exhaustive and reproducible evaluation of the aforementioned two state of the art methods on the Ottoman-Turkish makam recognition dataset.
4. Improvements in the state of the art for tonic identification in OTMM
5. The first neural network based algorithm for culture-independent mode recognition and its open source implementation.
6. Experiments on Hindustani and Carnatic music traditions to demonstrate the applicability of the implementations on different music cultures.

The rest of this paper is organized as the following: In Section 2 we provide the musical background relevant to these problems and methods. Section 3 provides a formal definition of the problem. Section 4 presents the current state-of-the-art. Section 6 describes the methods and their implementations in detail. Section 14 introduces the Ottoman-Turkish makam recognition dataset. Section 13 explains the experiments and the obtained results. Finally, we discuss the results we obtained in Section 19 and conclude with our comments and suggestions for the future work in Section 20.

2. MUSICAL BACKGROUND

3rd International Digital Libraries for Musicology workshop (DLfM) 2016 New York, USA

DOI: 10.475/123_4

define mode in "broad sense"? mode and tonic have synonymous meanings in many music cultures. tonal does not explain this concept. nasil cesitlidir? ornekler: makam, raag, raga, church modes etc. we should introduce what a stable pitch is. should we explain mono-/poly-/heterophony?

tonic as the reference for melody.

3. PROBLEM DEFINITION

We define *mode recognition* as classifying the mode $\zeta^{(a)}$ of an audio recording (a) from a discrete set of modes $Z = \{\zeta_1, \ldots, \zeta_V\}$, where $\zeta^{(a)} \in Z$ and V is the total number of modes. The mode set is specific to music culture being studied. In mode recognition, we assume that the tonic frequency $r^{(a)}$ of the audio recording is available.

We define *tonic identification* as estimating the frequency or the pitch class (if the octave information of the tonic is not be well-defined[1]), $r^{(a)}$ of the performance tonic. Unlike modes, frequency is a continuous variable. However, in practice, the tonic is typically constrained to be one of the stable pitches performed [10, 13]. With this assumption, tonic identification can be reformulated as estimating the tonic frequency or the pitch class $r^{(a)}$ from a finite set of stable frequency values $R = \{r_1, \ldots, r_W\}$ performed in an audio recording (a), where $r^{(a)} \in R$ and W is the number of the stable pitches in the audio recording. In tonic identification, we assume that the mode $\zeta^{(a)}$ of the recording is known.

A third scenario arises when both the tonic $r^{(a)}$ and the mode $\zeta^{(a)}$ of the recording (a) are unknown. In this case, we identify the tonic and recognize the mode simultaneously, which we term as *joint estimation*.

Note that these scenarios are actually multi-class problems, since the mode and the tonic may change throughout the performance. This is a more challenging problem, where we would not only like to obtain the set of the modes and tonics in the performance but also mark instances or intervals, where these changes take place.[2] As will be explained in Section 4, there has not been any generalizable method proposed for either mode recognition or tonic identification in such a scenario yet. In MORTY, we restrict the problem on the mode recognition and tonic identification of audio recordings with a single mode and tonic, leave the multiple estimation problem as a future work to investigate.

4. RELATED WORK

The use of pitch histograms haven't been limited to modal music domain, but extends to various music information retrieval tasks, such as key detection [27] and chord recognition [14]. However, these works deal with the equal-tempered euro-genetic music traditions and the used histograms have fixed size of 12 to capture the used 12 notes. This resolution is insufficient for representing the much denser frequency mappings of notes in certain modal music cultures [10, 13, 7].

[1]e.g. orchestral recordings of OTMM as explained in Section 13

[2]A manually annotated example for OTMM is given in http://musicbrainz.org/recording/37dd6a6a-4c19-4a86-886a-882840d59518

distribution-based methods asume modla smilarity, e.g. the shape of the distributions extracted from two different performances of the same mode should be similar. Musically, the shape similarity implies the tuning and the relative occurrence of the stable peaks are close to each each other.

which cultures need this

mode recognition - Koduri et al. reviews pitch distribution [17].

- transcription based modelsXX

Gopal use parameterized pitch distribution of individual svaras as features [18]

In [16] a method based on melodic contours is proposed for mode recognition. The method automaticaly extracts melodic phrases from audio recordings with labeled mode. These phrases indicate the characteristic movements of the performed mode. These phrases are then compared with melodic phrases automatically extracted from a test recording and the mode of the recording is estimated using a network analysis approach.

- dastgh musicXX - neural netsXX?

tonic identification state of the artXX why is it trivial? what are tradition specific approaches

To our knowledge, existing work on applications of neural networks on these problems have been limited to culture specific appraoches. In [26], the authors use neural networks for raga classification in Carnatic music tradition. However, they exploit a culture-specific heuristic which prevents this method to be extended to a culture-independent approach. They generate the pitch histogram of the recordings and use a 36-dimensional feature vector which consists of the frequencies of 12 highest peaks, their heights and variances. Notice that, the choice of 12 peaks correspond to 12 notes that might not be applicable to all modal music cultures. Another work [24] uses neural networks for raga recognition and uses arohana-avorahana sequence of the recording, which is a note sequence property specific to Carnatic tradition. Hence, this method isn't only ungeneralizable to arbitrary music traditions, but can't be apply to any other music tradition than Carnatic music. Moreover, neither provides the theoretical framework to demonstrate their method's performance in different configurations. Hence, our primary work on culture-independent applications of neural networks is a novel contribution and is intended to be a starting point for future work that is interested in modal music information retrieval using neural network approaches.

5. MORTY

Implementation details of MORTY. Licence, usage of open source Essentia library [6], numpy, scipy, sckitlearn [21], citeXX. The basic calls in a code environmentXX

6. k–NEAREST NEIGHBOR CLASSIFICATION

In MORTY we combine and generalize the two state of the art methods, originally proposed for audio recordings of OTMM [13] and short audio excerpts of Hindustani music [10]. The generalized method is supervised and use k-nearest neighbors (k-NN) for classification. Our implementation is generic such that the parameters selected in the feature extraction, training and testing steps can be optimized for the culture-specific properties of the studied music tradition. We also allow the user to classify either short audio

excerpts or complete audio recordings and switch between different features, training schemes and tasks as introduced in [13, 10]. Later in Section 13, we demonstrate the experiments for the parameter selection and optimization on a test dataset of OTMM (Section 14) and in Section 17 the reproduced results of [10] on a Hindustani and a Carnatic music dataset using the optimal parameters reported in [10].

In the training step we use audio excerpts with annotated mode and tonic. We first extract a predominant melody for each audio excerpt. These are used to compute either pitch distributions (PD) or pitch class distributions (PCD) (Section 7). Next, we create a model from the computed distributions (Section 8).

Given an audio recording with an unknown mode and/or tonic, we also extract the predominant melody and compute the distribution from the predominant melody. Then, we compute a distance or dissimilarity between the distribution of the test audio and the selected distributions in the training model, compute the k nearest neighbors from the computed measure (Section 9). Finally, we estimate the unknown mode and/or tonic as the most common label among the k nearest neighbors (Section 10-12).

Now we proceed to explain the generalized methodology in detail. We also label the input parameters of the implemented method in MORTY explicitly (e.g. **F1, P3**) throughout this Section for the sake of clarity.

7. FEATURE EXTRACTION

The first step of method is predominant melody extraction **(F1)** [10, 13, 7]. As discussed in [7] and [3], the quality of the extracted pitch predominant melody directly affects the reliability of the computed models and predominant melody extraction methods optimized or designed for the culture-specific aspects of the studied music might be desirable at this step. The implementation of such an algorithm is outside the context of MORTY.

We denote the predominant melody extracted from an audio excerpt, (a), as $X^{(a)} \triangleq \left(x_1^{(a)} \dots x_I^{(a)}\right)$, where $x_i^{(a)} \in X^{(a)}$ is a pitch sample and $i \in [1:I]$, where I is the length of the predominant melody.

Next, the samples in the predominant melody are converted to the cent scale using the equation below:

$$x^{(r)} \triangleq 1200 \log_2 \left(\frac{x}{r}\right) \tag{1}$$

Here, $x^{(r)}$ denotes the cent distance of the frequency x from the reference frequency r. In the training step (Section 8) and the mode recognition task (Section 10), i.e. when the annotated tonic is available, r is the annotated tonic frequency of the audio excerpt. In the tonic identification and joint identification tasks the predominant melody will be normalized with respect to the several tonic candidates one by one, one of which will be identified as the tonic (Section 11).

Using the normalized predominant samples we compute either a pitch distribution (PD) as used in [13] or a pitch class distribution (PCD) as used in [10]. PD and PCD shows the relative occurrence of the pitch and pitch class values with respect to each other, respectively. Throughout the text we simply refer the PDs and PCDs collectively as "distributions" **(F2)**. The values in both distributions are computed as:

$$h_n \triangleq \frac{\sum_{i=1}^{I} \lambda_n(x_i)}{I} \tag{2}$$

where h_n is the occurence computed for the n-th bin in the distribution h, computed samples $x_i \in X$ in the normalized pitch and I is the number of pitch samples.

The accumulator function λ for PDs is defined as:

$$\lambda_n(x) \triangleq \begin{cases} 1, & c_n \leq x \leq c_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where x is a normalized pitch sample and (c_n, c_{n+1}) are the bounds of the n-th bin. Similarly the λ function for PCDs is defined as:

$$\lambda_m(x) \triangleq \begin{cases} 1, & c_n \leq (x \mod 1200) \leq c_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Note that the PCD is a "circular" feature, e.g. the first and the last bins are next to each other. Also notice that both PD and PCD are normalized such that the resultant distribution is a probability density function.

The bin size β **(P1)** of the distribution determines how precise the distribution is (to the extend allowed by the cent-precision of the predominant melody) in representing the pitch space, the tuning of the stable pitches and the microtonal characteristics in a lower-level. The computed distributions might need to have a small bin size, e.g. less than a quarter tone (50 cents) for many music cultures [13, 10]. We select a constant bin size for the computed distributions, i.e. $\beta = c_{n+1} - c_n, \forall n$. The bin centers of both PDs and PCDs are selected such that the reference frequency r is represented as a bin centered around 0 cents. We denote the number of bins in a distribution as N. Note that N equals to $\lfloor 1200/\beta \rfloor$ in a PCD.

To remove the spurious peaks in the distribution we convolve it with a Gaussian kernel and obtain a "smoothed" distribution [10]. The standard deviation of the Gaussian kernel, termed as the kernel width σ **(P2)**, determines how smooth the resulting distribution will get. The kernel width should be comparable to the bin size **(P1)** since a value lower than one third of the bin size would not contribute much to smoothing[3] and a high value would "blur" the distribution too much. Moreover, this parameter has a direct impact on the number and the location of tonic candidates in tonic identification (Section 11), which might effect both the accuracy and the processing time. We select the overall width of the Gaussian kernel as 5 times the kernel width from peak to tail for performance reasons.

8. TRAINING MODEL

As mentioned earlier, the implemented method is supervised and hence require training data, i.e. audio excerpts with the annotated mode and tonic. From a training audio excerpt (a), we first extract the predominant melody $X^{(a)}$ and normalize with respect to the annotated tonic frequency $r^{(a)}$ (Equation 1). Next, the normalized predominant melodies $X^{(a,r^{(a)})}$ are grouped according to the annotated mode $\zeta^{(a)}$ of each individual except.

[3]Remember that the values of the bins in a Gaussian kernel, which are more than three standard deviations away from the mean are greatly diminished.

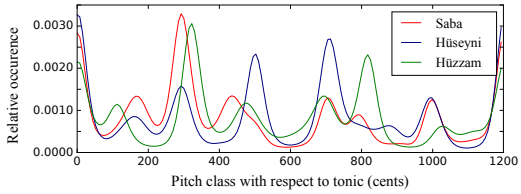

Figure 1: An example model with single PCD per mode trained for three makams

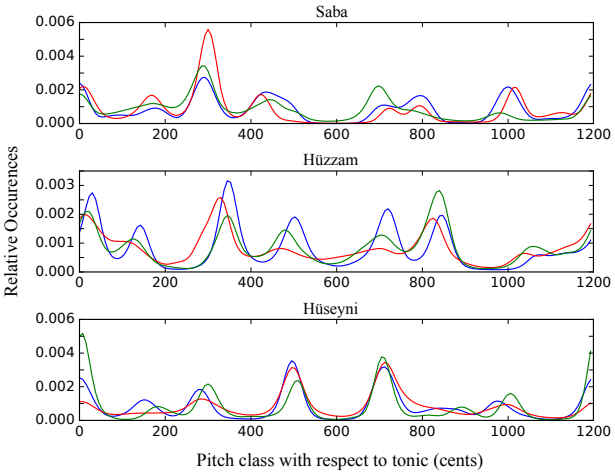

Figure 2: An example model with three PCDs per mode trained for three makams

The fundamental difference between the methods proposed in [13] and [10] is the training model **(M)** obtained in the training step. The methodology proposed by [13] joins all the normalized predominant melodies and compute a single PD or PCD per mode. [10] creates a separate distribution from each annotated excerpt (a). From a machine learning perspective [13] represents each mode with a single data point (Figure 1), whereas [10] represents them with many (Figure 2) in an N-dimensional space, where N is the number of bins in the distributions. From now on, we term the training models using the training step in [13] and [10] as "single distribution per mode" and "multi-distributions per mode", respectively. We denote the obtained model as $M \triangleq \{m_1, m_2, \dots\}$. $m_j \in M$ is a tuple $\langle h_j, \zeta_j \rangle$, where h_j and ζ_j denotes the trained distribution and the mode label of m_j, respectively. The model M consists of the distribution representations for V modes, where V is the number of unique mode labels $\zeta_v, v \in [1:V]$ in the training excerpts.

9. NEAREST NEIGHBOR SELECTION

In mode recognition, tonic identification and joint estimation tasks (Section 10-12), the basic step is to find the nearest neighbor(s) of a selected distribution among a set of distributions to be compared against. To find the nearest neighbors we compute a distance or a dissimilarity between the test distribution and each distribution in the comparison set [9]. We have currently implemented the distance and the similarity metrics in [13, 10], namely, City-Block (L_1 Norm) distance, Euclidean (L_2 Norm) distance, L_3 Norm, Bhattacharyya distance, intersection and cross correlation **(P3)**. Note that intersection and cross correlation are similarity metrics, hence we convert them to dissimilarities (i.e. 1−similarity) instead. The choice of the distance or dissimilarity measure plays a crucial role in the neighbor selection.

After the distances or the dissimilarities are computed, the compared distributions are ranked and the k **(P4)** nearest neighbors are selected. We then estimate the test sample as the most common label (ζ_v) of the neighbors. In case of a tie between two or more groups, we select label of the group, which accumulates the lowest distance or dissimilarity. Note that if a single-distribution is computed for each mode (**M** as explained in Section 8), the k value is always 1, since each mode is only represented by one sample.

Now we proceed to explain the procedure for each task **(T)** in detail.

10. MODE RECOGNITION

Given an audio excerpt (b) with an unknown mode, we compute the distribution $h^{(b,r^{(b)})}$ by taking the annotated tonic $r^{(b)}$ as the reference (Section 7). Next we compute the distance or the dissimilarity between $h^{(b,r^{(b)})}$ and the trained distribution h_j of each m_j, $\forall m_j \in M$, where M is the trained model, and obtain the k nearest neighbors to (b). We estimate the mode of (b) as the most common label ζ_v within the nearest neighbors as explained in Section 9.

11. TONIC IDENTIFICATION

Given an audio excerpt (b) with the annotated mode $\zeta^{(b)}$, we first extract the predominant melody X^b. Then we compute a distribution $h^{(b,*)}$ by taking an arbitrary frequency $(*)$ as the reference frequency (Section 7). We detect the peaks in the distribution using the method explained in [25]. The peaks indicate the stable pitches performed in the excerpt. We only consider the peaks, which have a ratio between its height and the maxima of the distribution above a constant threshold **(P5)**. We denote the set of tonic candidates as $R \triangleq \{r_1, \dots, r_W\}$, where W is the number of detected peaks. The cent distance between r_w and $*$ (Equation 1) is given as $r_w^{(*)} = (n_w - n_*) \times \beta, \forall l \in W$, where β denotes the bin size **(P1)** of the distribution **(F2)**, and n_w and n_* are the bin indices, in which l and $*$ reside in, i.e. $\lambda_{n_l}(l) = \lambda_{n_*}(*) = 1$ (Equation 2). Assuming each of the peaks r_w as the tonic candidate, we shift the distribution $h^{(b,*)}$ and obtain $h^{(b,r_w)}$ such that the n-th bin becomes the $(n+n_*-n_w)$-th for the PDs and the $(n+n_*-n_w \bmod K)$-th (where K is the total number of bins) for the PCDs, respectively and the n_w-th bin represents 0 cents in the shifted distribution.

From the training model M, we select all the m_j's $\in M$ with the label $\zeta^{(b)}$. Next we compute the distance or the dissimilarity between the each shifted distribution $h^{(b,r_w)}$ and the selected m_j. We select the k pairs with the lowest distance or dissimilarity and select the most occurring peak r_w in the neighbors as the estimated tonic (Section 9).

12. JOINT ESTIMATION

Given an audio excerpt (b) with unknown mode and tonic, we compute the tonic candidates, $R \triangleq \{r_1, \dots, r_W\}$ and the distributions $h^{(b,r_w)}$ assuming each $r_w \in R$ as the tonic candidate as explained in Section 11. Next we compute the distance or the dissimilarity between each shifted distribution $h^{(b,r_w)}$ and the m_j's in the training model M. We select

the k pairs with the lowest distance or dissimilarity and estimate the most occurring ⟨mode, tonic candidate⟩ pair, i.e. $\langle \zeta_v, r_w \rangle$ as the mode and the tonic (Section 9).

13. EXPERIMENTS

In this section, we provide the results of the experiments we did with the implementations provided in MORTY. We did exhaustive experiments using our dataset to demonstrate some properties of these methods, find the best parameter sets for OTMM and to provide some heuristics for future users. For the sake of reproducability all of the scripts, computed features, experiments and results are shared online[4].

14. TEST DATASET

In [13], the makam recognition methodology was evaluated on 172 solo audio recordings in 9 makams. To the best of our knowledge, this dataset represents the biggest number of recordings that has been used to evaluate makam recognition task, so far. As explained by the authors [13], these recordings were selected from the performances of "indisputable masters," and therefore they are musically representative of the covered makams. Nevertheless the results are not reproducible as the datasets are not provided by the authors.

The tonic identification methodology proposed by [13] was evaluated using 150 synthesized MIDI files plus 118 solo recordings. As explained in the above paragraph, the data is not available. To the best of our knowledge there exists only two open tonic identification datasets for OTMM, both of which are compiled under the CompMusic project.[5] The first one is provided in [23] and it consists of 257 audio recordings. The second and the bigger test dataset is provided in [2], consisting of 1093 recordings[6]. The recordings in both of the datasets are identified using MusicBrainz MBIDs[7]. The authors use a predominant melody extraction method proposed by [22] optimized for OTMM [3], which they have opened later[8]. Nevertheless, the predominant melodies extracted from the audio recordings are not provided in either dataset.

Considering the lack of open test datasets for makam recognition and the drawbacks of the available tonic identification datasets, we have gathered a test dataset of audio recordings with annotated makam and tonic, called the *Ottoman-Turkish makam recognition dataset*.[9] The dataset covers 20 commonly performed makams[10] and it is composed of 1000 audio recordings (i.e. 50 recordings per makam) in total. To the best of our knowledge, our dataset is the largest and the most comprehensive dataset for the evaluation of automatic makam recognition. Moreover, it is comparable to the dataset provided in [3] for the evaluation of tonic identification methodologies.

Similar to [23] and [2], the recordings in the dataset are labeled with MusicBrainz MBIDs. The tonic frequency of each recording is annotated manually using the procedure explained in [23] and the annotations are cross checked by at least two annotators. For reproducibility reasons, we also include the predominant melodies extracted from the audio recordings. We use the open implementation of the predominant melody extraction methodology in [3]. We additionally filter the predominant melody to get rid of the spurious estimations and correct the octave errors as explained in [7]. To the best of our knowledge this procedure [3] currently outputs the most reliable predominant melody estimations for OTMM.

Similar to [13], the dataset is intended to be musically representative of OTMM. To achieve this, we selected the recordings of acknowledged musicians from the CompMusic makam corpus, which is currently the most representative music corpus of OTMM aimed at computational research [28]. The dataset covers contemporary and historical, monophonic and heterophonic recordings, as well as live and studio recordings. Some of the recordings have non-musical sections, such as clapping at the end of live recordings, announcements in radio recordings or scratch and hissing sounds throughout the historical recordings. Pie chard of the instrument vocal etc.XX This diversity gives us the opportunity to test the methods in a much more challenging environment, which hasn't been completely addressed in the previous research [13]. Following our assumption in the problem definition, each recording belongs to a single makamXXrephrase.

15. EXPERIMENTAL SETUP

In the experiments we use stratified 10-fold cross validation. Table 1 gives a combination of the parameters used in the experimental setup. We use grid search, to find the optimal parameters for OTMM (Section 6). **(F1)** is selected as the state of the art in predominant melody extraction for OTMM. The parameter combinations where the bin size β is greater than or equal to 3 times the kernel width σ are omitted. We also conduct experiments where the distribution smoothing is skipped. When the training model consists of a "single" distribution per mode, the number of neighbors k is always taken as 1 as each label is represented by a single data point. The minimum peak ratio **(P5)** is only used in tonic identification and its optimal value is found separately as will be explained in Section 16.

For mode recognition, we mark the classification is $True$ if the estimated mode and the annotated mode for a recording are the same. The tonic octave in the orchestral performances of OTMM is ambiguous as each instrument plays the melody in their own register. Therefore, we aim identify the tonic pitch class. We calculate the octave wrapped cent distance between the estimated and the annotated tonic, i.e. $min\left(\left(|e^{(r)}| \bmod 1200\right), 1200 - \left(|e^{(r)}| \bmod 1200\right)\right)$, where mod is the modulo operation. Remember that $e^{(r)}$ is the normalization of the estimated tonic frequency e, with respect to the annotated tonic frequency r (Equation 2). If the cent distance is less than 25, we consider the tonic identification as correct. In the case of joint estimation, we require both tonic and mode estimates to be correct.

[4] https://github.com/sertansenturk/dlfm_makam_recognition

[5] http://compmusic.upf.edu/

[6] The datasets are hosted in https://github.com/MTG/turkish_makam_tonic_dataset/releases/

[7] https://musicbrainz.org/doc/MusicBrainz_Identifier

[8] https://github.com/sertansenturk/predominantmelodymakam

[9] https://github.com/MTG/ottoman_turkish_makam_recognition_dataset

[10] i.e. Acemaşiran, Acemkürdi, Bestenigar, Beyati, Hicaz, Hicazkar, Hüseyni, Hüzzam, Karcığar, Kürdilihicazkar, Mahur, Muhayyer, Neva, Nihavent, Rast, Saba, Segah, Sultanıyegah, Suzinak and Uşşak

Table 1: The summary of the parameters tried in the experiments

	Name	Values	Comment
T	task	mode, tonic, joint	
F1	predominant melody, X	[3]	extraction method specialized for OTMM
F2	distribution, h	PD, PCD	
M	type of the training model, M	single, multi	number of distribution per mode used in [10, 13]
P1	bin size, β	$7.5, 15, 25, 50, 100$ cents	
P2	kernel width, σ	"no smoothing & $7.5, 15, 25, 50, 100$ cents"	Combinations with $\beta \geq 3\sigma$ are omitted.
P3	distance or dissimilarity	L_1, L_2, L_3, Bhattacharyya, 1−intersection, 1−cross_correlation	1−intersection and 1−cross_correlation are dissimilarities computed from the namesake similarity measures
P4	number of nearest neighbors, k	$\{1, 3, 5, 10, 15\}$	for the "single" distribution per mode training model, the value is fixed to 1
P5	minimum peak ratio	$[0, 1]$	with a step of 0.5, only for tonic identification and joint estimation

For each fold we compute the accuracy, which is the number of correct estimations divided by the total number of testing data. In Section 16, we report the average accuracies of the folds for each parameter combination. For all results below, the term "significant" has the following means that the claim is statistically significant at the $p = 0.01$ level as determined by a multiple comparison test using the Tukey-Kramer statistic.

We additionally compare the tonic identification results obtained in the tonic identification and joint estimation tasks with the results obtained from the current state of the art in OTMM [2]. The method is based on detecting the last stable pitch of the recording, which is typically the tonic.[11].

Table 2: Best parameter sets for each task and method.

Task	Method	sf	ss	k	Accuracy
Tonic	Multi	15	15	3 or 5	**94.9%**
		15	7.5	5	
		0	7.5	3	
	Single	20	7.5	-	94.5%
Mode	Multi	20	25	5	70.6%
	Single	15	7.5	-	**71.4%**
Joint	Multi	20	15	5	62.1%
	Single	15	7.5	-	**63.2%**

16. RESULTS

Minimum peak ratio experiment XX

Table 16 shows the best results obtained after grid search. For mode recognition, multiple distribution per mode model yields an accuracy of 70.6% with the best parameter set and the single distribution per mode method yileds a slightly higher accuracy of 71.4%. For tonic identification multiple distribution per mode performs with above 94% accuracy in 49 sets and above 90% accuracy in 180 parameter sets out of 540 experiments, where the highest accuracy obtained is 94.9%. Hence, the method is robust to a variety of parameter selections for tonic identification. Single distribution per mode model performs on par with the multi-model approach and yields 94.5% accuracy with the best parameter set. The method proposed by [2] performs with 79.9% accuracy in tonic identification on our dataset. For joint estimation the distribution per mode model performed with 63.2% accuracy in the optimal configuration while multi-distribution yielded 62.1%.

In both methods and all three tasks, the distance method of all best parameter sets turned out to be Bhattacharyya distance and the histogram type to be PCD.

These experiments revealed that certain parameter selections significantly improve or diminish the methods' performances. These observations are listed below as heuristics for future users.

- Single-histogram mode models approach perform slightly better than multi-histogram models. The difference is insignificant.
- PCD significantly outperforms PD.
- Bhattacharyya, intersection and city-block distance methods significantly outperform cross-correlation, L_3 and Euclidean.
- Smaller bin size yield better results, however there is no significant between 7.5, 15 and 25 cent bin sizes. Note that these bin sizes significantly outperform 50 and 100 cent bin sizes.
- Increasing smoothing above 7.5 cents improves the accuracy of the models, but this change is not significant.XX
- In the case of multi-distribution per mode model, the value of k does not make a significant impact, however a k value of 5 performs better.XX

Confusion matrix XX

17. EXPERIMENTS ON HINDUSTANI AND CARNATIC MUSIC

Recently, MORTY had been used as a benchmark for raga/raag recognition of audio recordings of Hindustani and Carnatic music in comparison with two novel methods [16, 1]. In the experiments we use the optimal parameters reported in [10]: We compute pitch class distributions with a bin size of $\beta = 10$ cents. The kernel width σ reported in [10] is in Hz scale, which is would mean variable smoothing across octave. We empirically found $\sigma = 15$ cents as a reasonable value. We train multiple pitch class distributions per mode as described in [10]. In testing we use Bhattacharyya distance and select the nearest neighbor ($k = 1$). Note that there already exists a method for tonic identification for these music traditions [15], which is reported to provide near perfect results. We use this method for the

[11]The open implementation is available at https://github.com/hsercanatli/tonicidentifier_makam

automatic tonic identification instead of the methodology explained in Section 8. Below we explain the results briefly. Please refer to the respective papers [16, 1] for further details.

In [16] a method based on melodic contours is proposed for mode recognition. The method is compared against the methods proposed in [10] (our method) and [18] on two datasets consisting Carnatic music recordings in 10 ragas and 40 ragas, respectively. The best accuracies obtained for the 10 raga dataset are 91.7%, 89.5% and 70.01% using the proposed method, MORTY and [18]. Both the proposed methodology and MORTY obtained comparable results and they outperformed [18]. In the 40 raga dataset our method (74.1% accuracy) outperformed the both methods the proposed method (69.6% accuracy) and [18] (51.4% accuracy).

Recently the same authors have proposed another feature for raga recognition [1], which they term as "time-delayed melody surface." The training and testing scheme is the same with the procedure explained in [10]. In the experiments The proposed feature was compared against pitch class distributions on a dataset of Hindustani recordings and the same dataset used in [16] with a different experimental setup. In both datasets the proposed feature (97.7% accuracy on Hindustani and 86.7% accuracy on Carnatic dataset) has outperforms PCDs (91.7% accuracy on Hindustani and 73.1% accuracy on Carnatic dataset).

18. MULTI-LAYER PERCEPTRON NETWORK

Multi-layer perceptron network (MLP) or fully connected neural network [19] is one of the most commonly used general purpose neural network architectures in the literature. As a primer work on applying neural networks on culture-independent modal music information retrieval, we decided to use this architecture, due to its established theory and convenience for theoretical computations.

In different contexts, MLPs are shown to perform substantially better as the amount of training data is increased [11]. Having curated the largest OTMM recording dataset in the literature, we had already overcome a challenging prerequisite for using neural networks for these task, considering the lack of available public datasets for the domain.

As described in more detail in the following section, this dataset is composed of 1000 recordings, distributed as 50 recording for each of 20 most common makams in CompMusic corpus [28]. Hence, we define our problem as 20-class mode recognition with a training dataset of 1000 samples.

Since the complexity of the network and hence the data required to train it is proportional to the number of nodes in the network [29] [4], we decided to use PCD as the network's input feature, instead of PD. In addition to its ability to represent the recordings more compactly, the fixed size of PCD make it a much more convenient choice than the variable length and variable range alternative PD. With the same motivation, we decided to use a MLP with a relatively small size, with no hidden layers, input layer of size equal to the size of PCD and an output of size equal to number of classes. The output of each output neuron would be treated as the confidence of the network of the input belonging to the corresponding class. The highest confidence, in other words the index of maximum value from the output layer, is treated as the network's estimate of mode.

One of the most challenging steps of theoretical studies on neural networks, is representing the complexity of the network, which can be quantified by the size of the network's hypothesis set, H. The hypothesis set is a function space which spans the possible functions the neural network might converge to, through training. Since it is very hard to estimate this parameter, it is common to work with growth functions, denoted by $m_H(N)$. It refers to the growth of the hypothesis set with the increase of number of samples, denoted here with N. More concretely, $m_H(N)$ denotes the most number of dichotomies that H implements over all possible training sets of size N. One way to work with this value is to use VC-dimension [29], denoted by d_{VC}, as an upper bound on it. As a heuristic, d_{VC} is approximated as the number of parameters [4] for MLP. We use this approach and the following theorem [12] to obtain a lower bound on the required number of samples to be able to train the network we specified fully.

Theorem [12]: Let L be a learning algorithm that uses H consistently. Then, given $0 < \epsilon < \frac{1}{8}$ and $0 < \delta < \frac{1}{100}$ and less than N random examples, there is some probability distribution that for which L will not produce a function $h \in H$, with $error(h) \leq \epsilon$ with probability $1 - \delta$, where:

$$N = \frac{d_{VC}(H) - 1}{32\epsilon} \tag{5}$$

This theorem provides us a lower bound on the number of training samples to ensure that the learning algorithm is trained enough to be able to converge to all functions in its hypothesis set with error less than ϵ with probability $1 - \delta$. When this lower bound for our non-complex network is computed, it shows that even the largest dataset in the literature is far from being enough to ensure this condition.

As an example, we consider using 160 dimensional and 80 dimensional PCDs, which correspond to 7.5 cents and 15 cents bin sizes and are typical choices for mode and tonic estimation tasks [10, 13, 7]. We chose the error margin as $\epsilon = 0.1$ and calculated the lower bounds of training set sizes as 9100 and 2550, respectively. Since we only have 1000 recordings available, we don't expect our neural network implementation to be a reliable choice for these tasks, with this configuration.

To tackle this problem, we considered using principal component analysis (PCA) [20] for dimension reduction, which would diminish the size of the network while preserving the variance among training samples. However, our experiments on our dataset yielded unsatisfactory results.

19. DISCUSSION

The drawback of the pitch class distribution method is that it does not consider the temporal characteristics. When we inspected the results obtained from the experiments in 13, we observed that the confusions are mainly between makams, which either have very similar intervals in their scale. Similarly in [16], the proposed method was better in classifying phrase-based ragas, while our method was better at classifying scale based ones. The mode recognition using the feature proposed in [1] is able to capture both of these properties better with a slight increase in computational complexity.

Nevertheless, [2] yields better results in tonic identification when the makam information is not available.

Therefore, our dataset can be used complementary with the earlier tonic datasets [23, 3]. Our method outperforms [2], if the makam of the recording is known.

In the light of these information and our empirical observations, we suggest using single-histogram models approach with Bhattacharyya distance, PCD, smoothing factor of 20 cents and bin size of 7.5 cents for a system that performs well in all of the three tasks.

20. CONCLUSION

This primary research can be used as a starting point for future works that aim to use neural networks for modal music information retrieval tasks.

Out of the listed methods that make use of pitch histograms, we decided to focus on single histogram mode models [13, 7] and multi-histogram mode models [10] approaches because the first is the state of the art for mode and tonic recognition tasks in OTMM culture and the latter is for joint recognition for Hindustani, while being on par with the state-of-the-art [16] for mode recognition in Carnatic tradition. In addition to these existing methods, we implemented a neural network model for culture-independent multi-mode recognition. The recent vast improvements in neural network methods and the availability of data due to our dataset makes this method a promising candidate for future research on this problem.

21. REFERENCES

[1] Anonymous. Reference suppressed for anonymity.

[2] H. S. Atlı, B. Bozkurt, and S. Şentürk. A method for tonic frequency identification of Turkish makam music recordings. In *5th International Workshop on Folk Music Analysis (FMA)*, pages 119–122, Paris, France, 2015.

[3] H. S. Atlı, B. Uyar, S. Şentürk, B. Bozkurt, and X. Serra. Audio feature extraction for exploring turkish makam music. In *3rd International Conference on Audio Technologies for Music and Media*, Ankara, Turkey, 2014. Bilkent University.

[4] P. L. Bartlett, V. Maiorov, and R. Meir. Almost linear VC-dimension bounds for piecewise polynomial networks. *Neural Computing*, 10(9):2159–2173, 1998.

[5] E. Benetos and A. Holzapfel. Automatic transcription of Turkish microtonal music. *Journal of the Acoustical Society of America*, 138(4):2118–2130, 2015.

[6] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[7] B. Bozkurt. An automatic pitch analysis method for Turkish maqam music. *Journal of New Music Research*, 37(1):1–13, 2008.

[8] B. Bozkurt. A system for tuning instruments using recorded music instead of theory-based frequency presets. *Computer Music Journal*, 36:43–56, 2012.

[9] S.-H. Cha and S. N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370, 2002.

[10] P. Chordia and S. Şentürk. Joint recognition of raag and tonic in north Indian music. *Computer Music Journal*, 37(3):82–98, 2013.

[11] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010.

[12] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:247–261, 1989.

[13] A. C. Gedik and B. Bozkurt. Pitch-frequency histogram-based music information retrieval for Turkish music. *Signal Processing*, 90(4):1049–1063, 2010.

[14] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.

[15] S. Gulati, A. Bellur, J. Salamon, H. G. Ranjani, V. Ishwar, H. Murthy, and X. Serra. Automatic tonic identification in Indian art music: Approaches and evaluation. *Journal of New Music Research*, 43:53–71, 2014.

[16] S. Gulati, J. Serrà, V. Ishwar, S. Şentürk, and X. Serra. Phrase-based rāga recognition using vector space modeling. In *41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, pages 66–70. IEEE, 20/3/2016 2016.

[17] G. K. Koduri, S. Gulati, P. Rao, and X. Serra. Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4):337–350, 2012.

[18] G. K. Koduri, V. Ishwar, J. Serrà, and X. Serra. Intonation analysis of rāgas in Carnatic music. *Journal of New Music Research*, 43(01):72–93, Jan. 2014.

[19] C. Malsburg. *Brain Theory: Proceedings of the First Trieste Meeting on Brain Theory, October 1–4, 1984*. Springer Berlin Heidelberg, 1986.

[20] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572, 1901.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

[23] S. Şentürk, S. Gulati, and X. Serra. Score informed tonic identification for makam music of Turkey. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, pages 175–180, Curitiba, Brazil, 2013.

[24] S. Shetty and K. Achary. Raga mining of Indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, 1:362–366, 2009.

[25] J. O. Smith III and X. Serra. *PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation*. CCRMA, Department of Music, Stanford University, 1987.

[26] S. M. Suma and S. G. Koolagudi. *Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, Volume 1*, chapter Raga Classification for Carnatic Music,

pages 865–875. Springer India, New Delhi, 2015.

[27] D. Temperley and E. W. Marvin. Pitch-class distribution and the identification of key. *Music Perception: An Interdisciplinary Journal*, 25(3):193–212, 2008.

[28] B. Uyar, H. S. Atlı, S. Şentürk, B. Bozkurt, and X. Serra. A corpus for computational research of Turkish makam music. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, pages 1–7, 2014.

[29] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.