Recognizing Ornaments in Vocal Indian Art Music with Active Annotation

Sumit Kumar†, Parampreet Singh†, Vipul Arora§
†Graduate Student Member, IEEE, §Senior Member, IEEE
†§Department of Electrical Engineering, Indian Institute of Technology, Kanpur
§Department of Electrical Engineering (ESAT), KU Leuven

***Abstract*—Ornamentations, embellishments, or microtonal inflections are essential to melodic expression across many musical traditions, adding depth, nuance, and emotional impact to performances. Recognizing ornamentations in singing voices is key to MIR, with potential applications in music pedagogy, singer identification, genre classification, and controlled singing voice generation. However, the lack of annotated datasets and specialized modeling approaches remains a major obstacle for progress in this research area. In this work, we introduce Rāga Ornamentation Detection (ROD), a novel dataset comprising Indian classical music recordings curated by expert musicians. The dataset is annotated using a custom Human-in-the-Loop tool for six vocal ornaments labelled as events with start and end timestamps. Using this dataset, we develop an ornamentation detection model based on deep time-series analysis, preserving ornament boundaries during the chunking of long audio recordings. We conduct experiments using different train-test configurations within the ROD dataset and also evaluate our approach on a separate, manually annotated dataset of live concert recordings. Experimental results support the superior performance of our proposed approach over the baseline CRNN model.**

***Index Terms*—Ornamentation detection, Singing techniques, MIR, Temporal Convolutional Networks, Indian Art Music**

I. Introduction

MUSICAL ornaments are intentional variations around notes that bring expressiveness and complexity to a melody [1], [2]. They are essential elements across musical traditions, shaping the music's identity and emotional depth. Automatic detection of ornaments in any music piece has broad applications in music education, performance analysis, and computational musicology, yet it remains significantly understudied, with very few works like [3]–[5].

The task of ornamentation detection is typically formulated as an Audio Event Detection (AED) problem [3], [6], and therefore, several AED strategies and baseline models such as CRNN [7], [8] are commonly employed. However, this direct application presents certain challenges. Unlike standard audio events (e.g., clapping, car honks) that remain identifiable even when partially captured, musical ornaments require their full temporal structure for accurate classification. The common practice of segmenting audio into fixed-length chunks during pre-processing can fragment the ornaments across boundaries. For instance, an ornament split between two chunks with one half at the end of one clip and the other at the beginning of the next would lose its structural integrity, leading to fragmented labels and misclassification during training. Furthermore, effective progress in this area requires carefully curated datasets, specifically annotated to mark precise event boundaries of ornaments present within a musical piece. This level of annotation requires time-intensive efforts, domain expertise, and trained musicians, along with a structured annotation platform, making it an expensive, labor-intensive, and non-trivial task to undertake.

Analysis and detection of vocal ornamentations are highly relevant for both academic research and practical applications. One of the primary applications is in music pedagogy [9]–[11], where ornamentation detection can assess a student's accuracy to replicate expressive nuances and provide focused feedback to improve vocal performance. In controlled singing voice or melody generation [12]–[14] , ornamentation detection enables synthesis models to replicate expert singing styles more naturally by conditioning on detected ornamentation patterns. For singer identification, analyzing a singer's distinctive use of ornamentations can enable recognition systems to differentiate vocalists based on their unique stylistic patterns [15].

In this work, we present a comprehensive study on ornamentation detection in Indian Art Music (IAM). We introduce the Rāga Ornamentation Detection (ROD) dataset, specifically curated for detecting ornaments in vocal performances, with detailed frame-level annotations for six distinct ornamentation types. To streamline the annotation process, we design a custom annotation tool incorporating a human-in-the-loop (HITL) framework within an active learning setting. To address label fragmentation during chunking, we propose a don't care labelling-based training strategy. Using the ROD dataset, we train a deep learning model based on Temporal Convolutional Networks (TCNs) and demonstrate its performance over the baseline CRNN model. We experiment with various train-test split scenarios across different *rāgās* and singers to assess the model's robustness, and further evaluate its generalizability using a separate dataset of real-world IAM concert performances taken from Prasar Bharati[1] Archives [16]. We then conduct ablation studies on our TCN model, highlighting the importance of 'don't care' labelling, periodic padding, and dilated convolutions on our proposed model. Our codes, dataset, pre-trained models, and demos are available on https://github.com/madhavlab/2024_ornamentation.

[1]Prasar Bharati is India's public broadcasting agency, comprising Doordarshan Television Network and All India Radio. It maintains an extensive archive of Indian classical music recordings.

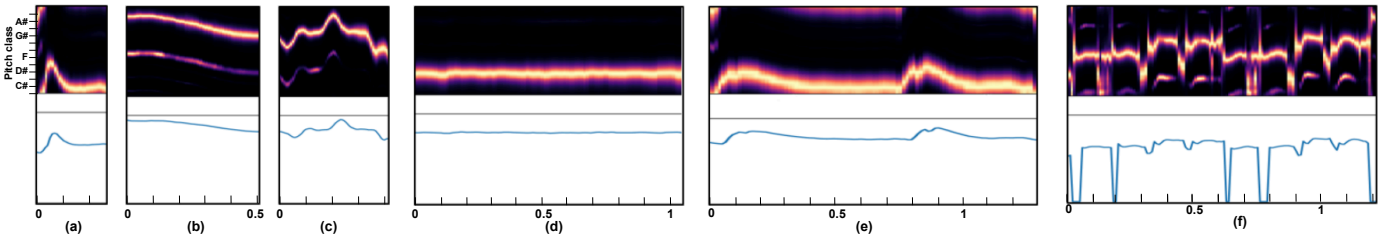


Fig. 1. Chromagram (top) and Pitch Contour (bottom) representations of ornaments in IAM (a): *Kaṇ*, (b): *Mīnd*, (c): *Murkī*, (d): *Nyās svar*, (e): *Andolan*, and (f): *Gamak* in ROD dataset. x-axis represents time (in sec.)

II. Ornaments in Indian Art Music

Derived from the *Saṁskṛita* word meaning "beautification", *Alaṅkāras* or ornaments constitute any pattern of musical decoration created by musicians within or across tones [17]. In IAM, ornaments are essential to accurate *rāga* rendition, whereas in Western music, they are often optional embellishments [5], [5]. The six significant ornaments in IAM [5], [17], [18] can be seen in Figure 1 are:

1) Kaṇ (grace notes): *Kaṇ* or the grace note is a subtle, ornamental note in HCM (Hindustani classical music) that briefly touches the main note, either from above or below, without full articulation. It is rendered swiftly and delicately, enhancing melodic fluidity and expressiveness.

2) Mīnd (gliding between notes): In *Mīnd*, or glide, the pitch smoothly transitions between two notes, creating a continuous, flowing effect. Unlike *Kaṇ*, which is a brief and subtle touch of an adjacent note, *Mīnd* involves a sustained and gradual pitch movement, often spanning multiple notes. *Mīnd* can be thought of as an elongated and more fluid counterpart to *Kaṇ*.

3) Murkī (subtle renditions of note clusters): A *Murkī* is similar to a trill, typically involving the rapid and light alternation of two or three neighbouring notes. In fast-paced, folk-derived compositions, *Murkīs* are executed to sound light and sharp. In slower, more sensuous compositions, they are performed more smoothly, creating a languid rather than sharp effect. Due to the flexibility inherent in its execution, *Murkī* poses a significant challenge for automatic recognition and detection.

4) Nyās svar (emphasized resting notes): *Nyās svar,* or holding notes are specific notes within a *rāga* that can be sustained for extended durations. *Nyās svar* is the note on which a *rāga* concludes.

5) Andolan (gentle oscillations): An *Andolan* is a gentle swing or oscillation that begins from a fixed note and reaches the periphery of another note. During these oscillations, it touches various microtones present between the notes. *Andolan* can be conceptualized as multiple *Mīnd* in succession, punctuated by delicate holds that define its expressive motion.

6) Gamak (heavy oscillations): A *Gamak* is a rapid oscillation between two notes, performed with deliberate force and vigour. Compared to *Murkī*, *Gamakās* are generally more intense, faster, and executed with greater force, often lasting for a longer duration.

III. Related Works

A. Datasets for ornamentation detection task

Phonation Mode dataset [19] consists of one-second recordings annotated at the clip level, covering four vocal modes, namely: neutral, pressed, breathy, and flow, produced by four different singers to detect phonation modes in sustained sung vowels. While it includes a broad pitch range, the discrete nature of the pitches lacks melodic context. Similarly, the PMSing dataset [20] extends phonation mode detection by incorporating adversarial discriminative training to improve classification accuracy. However, both datasets focus on phonation modes rather than detailed ornamentation.

VocalSet [21] addresses this limitation by incorporating vocal performances in scales, arpeggios, long tones, and musical excerpts. It includes a diverse range of singing techniques, such as vibrato, trills, vocal fry, and inhaled singing. However, the dataset lacks strong labelling, as each audio file contains only one type of ornamentation, differing significantly from real-world music recordings. Moreover, several datasets have been annotated for singing voices in real-world settings [3], [22]–[24]. The KVT dataset [22], initially developed for singing voice tagging in K-POP, includes 70 vocal tags, with six related to singing techniques, such as whisper/quiet, vibrato, shouty, falsetto, speech-like, and non-breathy. However, it is not specifically designed for singing technique detection. The dataset includes annotations at both the song level and 10-second segment level, making it suitable for broad categorization but less detailed in ornamentation analysis. Similarly, the Singing Voice Query by Tag Dataset (SVQTD) [23] focuses on singing voice retrieval using descriptive tags, including timbral and phonation-based features. While it provides rich metadata for vocal characterizations, it lacks detailed temporal annotations, i.e., strong labels necessary for fine-grained ornamentation detection.

Further, the MVD dataset [24], designed for analyzing screams in heavy metal music, provides strong labelling for four types of screams—high fry, mid fry, low fry, and layered. Although it includes precise annotations, it was developed for scream detection rather than general singing techniques. Furthermore, the authors in [3] present a dataset that consists of 168 commercial J-POP songs, with annotations of various singing techniques marked with precise timestamps and corresponding vocal pitch contours. However, as the audio files are commercially licensed and only available through platforms such as Apple iTunes, the lack of open access poses challenges to reproducibility and may hinder further research.

Moreover, the work [18] explored computational analysis of ornamentations for evaluating singing quality, marking another step towards objective modeling of expressive elements in Indian classical music.

In contrast, our proposed ROD dataset is substantially larger and more diverse, with frame-level event annotations for six ornament types.

Singing techniques are deeply rooted in the cultural and stylistic characteristics of a particular genre, making generalization across traditions challenging. Therefore, to address the task of ornamentation detection in IAM, we curate the ROD dataset, which we describe in detail in Section IV.

B. Analysis and Detection of Events

The detection and analysis of events have been a long-standing area of research in audio and video signal processing [25], [26]. In the domain of audio event detection (AED), various methodologies have been proposed to automatically identify and classify acoustic events in diverse contexts, ranging from environmental sounds [27]–[30], to structured musical content like percussion stroke detection, drum transcription, etc. [31], [32]. Similarly, in video signal processing, action segmentation techniques aim to segment and recognize temporal sequences of human activities using deep learning-based architectures such as Temporal Convolutional Networks (TCNs) [33]–[35].

Given the conceptual overlap between general event detection tasks and ornamentation detection in music, we approach the latter as a specialized case of AED. The work [36] uses a small dataset to detect a limited set of ornaments (*Mīnd*, *Andolan*, and holding notes) using pitch-contour fitting and rule-based template matching, while relying on handcrafted thresholds. Building upon such early efforts, the baseline study for this task [3] adopts a conventional Convolutional Recurrent Neural Network (CRNN) architecture, formulating ornamentation detection as a frame-wise classification problem. In contrast, we employ a Temporal Convolutional Network (TCN)-based approach, which is discussed in detail in Section V-B.

IV. Dataset Description

The Rāga Ornamentation Detection (ROD) dataset is a curated collection of Hindustani Classical Music (HCM)[2] vocal performances for the study of ornamentations in IAM. It consists of 212 audio files recorded by two expert singers, totaling 4.08 hours of audio. Singer 1 and singer 2 contributed 108 and 104 recordings, respectively. Each audio file is a mono-channel WAV file recorded at 44.1 kHz sampling rate and 32-bit precision, using an Audio-Technica AT2020 condenser microphone and a laptop with a 12-core Intel Core i5-12500H processor. The dataset spans four distinct *rāgas* across the two singers, with details summarized in Table I. While recording, the singers were given the flexibility to design structured lessons, maintaining that all exercises within a lesson adhered to a single *rāga*. Lessons were organized to progress from easy to difficult levels, targeting pedagogical objectives. During recordings, tanpura and tabla accompaniments were played through headphones for tonal and rhythmic reference, while only the vocal track was recorded as a clean mono signal. Metadata, including tonic, *tāla*, tempo (BPM), and accompaniment files, is provided to allow later merging of accompaniment and vocal tracks if needed.

[2]Hindustani Classical Music is a branch of Indian Art Music popular in Northern regions of India.

A. Annotation Process

The annotation process follows a two-stage approach by two sets of expert annotators. In the first stage, two annotators, each being the vocalist of the respective recordings, manually annotate their audio files by listening to them. Using Audacity [37] software, they mark the onset (start time), offset (end time), and the type of ornamentation (class) for each occurrence. The ornaments are strongly labelled as non-overlapping events, similar to those in Audio Event Detection (AED) tasks. Since labelling musical ornamentations differs significantly from labelling typical audio events in AED tasks [30], [38], subjectivity and stylistic variations between annotators can introduce discrepancies in the first round of annotation. Annotators may mark different labels for similar patterns or merge two closely occurring ornamentations into a single label, leading to variations in event segmentation.

To refine the annotations, a second stage is carried out by two additional annotators using a model trained on the first-round labels. These annotators are given the same audio recordings along with a Tkinter-based UI (as shown in the Supplemental Material) that displays chromagram visualizations, pitch contours, first-stage annotations, and the model's predictions. This setup, in an active learning framework, allows them to make precise adjustments to event timings. The second round of annotation addresses inconsistencies from the first round by refining event boundaries and ensuring that closely occurring ornamentations are distinctly labelled. For instance, in cases where multiple *Kaṇ* appear in quick succession, they are individually marked rather than being merged into a single event.

Figure 2 shows ornamentation frequency observed in the performances of two vocal singers. Singer-1 demonstrates higher usage across most categories. This variation can be attributed to differences in their musical *gharānās* [39] (lineage-based musical tradition of IAM), which influences ornamentation and singing styles in general. The duration distribution of each ornament for both singers is shown in Supplemental Material (Figure: S3, S4).

1) Ornamentation-Specific Annotation Rules: To standardize the labelling process, specific duration constraints and clear event boundaries are established for each ornamentation type. For *Kaṇ*, a maximum duration is limited to 0.35 seconds to avoid confusion with *Mīnd*. In the case of *Mīnd*, a minimum duration of 0.45 seconds is required, and only the gliding portion is labelled, excluding sustained holds after the glide. For *Nyās Svar*, a minimum duration of 0.6 seconds is set without imposing an upper limit. *Murkī* is constrained to a duration range of 0.4 to 1.0 seconds. For *Andolan*, a minimum

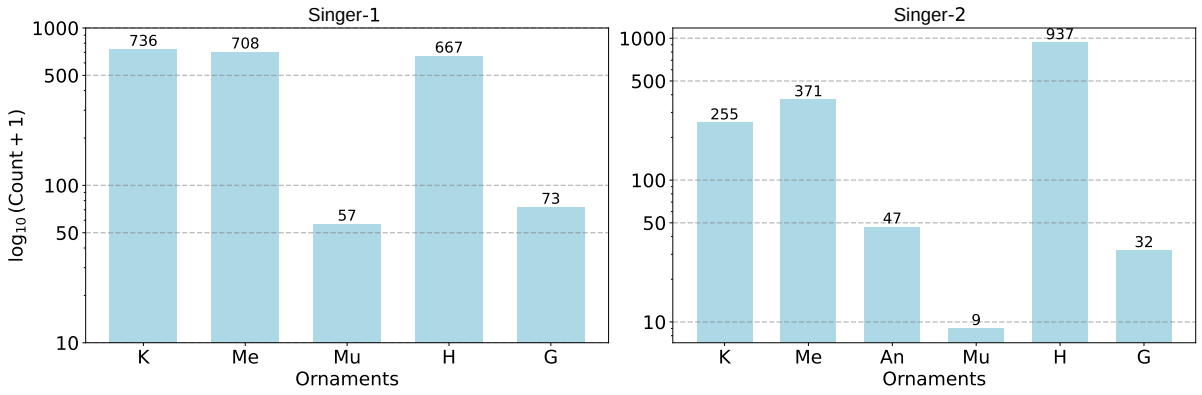


Fig. 2. Class-wise ornament frequency for Singer 1 (left) and Singer 2 (right), with ornament classes on the x-axis and log-scaled counts on the y-axis. K, Me, An, Mu, H, and G represent *Kaṇ*, *Mīnd*, *Andolan*, *Murki*, *Nyās Svar*, and *Gamak* respectively.

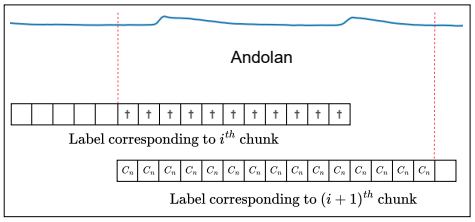


Fig. 3. An illustration of audio chunking using overlapping windows and corresponding label encoding. Label C_n corresponds to class *Andolan* (within red lines) and † is the 'don't care' label.

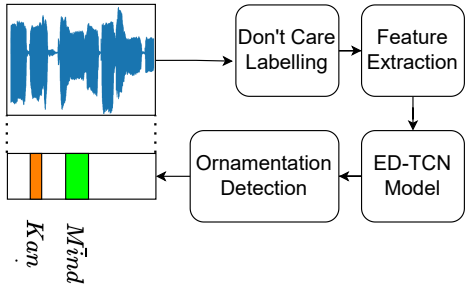


Fig. 4. An overview of the proposed methodology for the ornamentation detection task

duration of 1.0 seconds is required to distinguish it from isolated *Mīnd*. Lastly, *Gamak* is annotated only if it persisted for at least 0.7 seconds to avoid misclassification with *Kaṇ* or *Murkī*.

B. Inter-Annotator Agreement

To account for subjectivity that might be seen in the annotation because of differences in expertise, perception, stylistic variations, and interpretations across *Gharānās*, we assess inter-annotator agreement (IAA) using Cohen's Kappa score. An additional set of annotators, using the rules and annotation setup, independently perform a blind round of annotation for both datasets, resulting in a Cohen's Kappa score of 0.79.

TABLE I
DURATION-WISE DISTRIBUTION OF AUDIO RECORDINGS ACROSS DIFFERENT RĀGAS AND SINGERS IN THE ROD DATASET.

Singer	*Rāga*	Duration (in seconds)
Singer 1	*Bāgeshrī*	2188
	Bhūpāli	1911
Singer 2	*Bhairav*	3345
	Bhūpāli	6967
	Darbārī	282

V. Proposed Methodology

We formulate the ornamentation detection task in analogy with audio event detection (AED), while introducing critical adaptations tailored to the unique characteristics of ornamentations. The proposed methodology comprises several stages, including data pre-processing, feature extraction, and the design of a deep learning model as shown in Figure 4. Unlike conventional AED approaches, our pipeline incorporates modifications to preserve the temporal structure of ornamentations during both pre-processing and model training, ensuring accurate temporal localization and characterization of ornaments.

A. Data Preprocessing

In general, audio recordings in a dataset vary in length [40]–[42]. To use audio or extracted features as input to a deep learning model, the recordings are generally segmented into fixed-duration chunks [3]. However, in our task, this chunking strategy can adversely affect ornaments occurring at the boundaries of segments. For instance, an *Andolan* that is abruptly segmented may resemble a *Mīnd* or a short portion of *Gamak* may resemble *Murkī*. To address this issue, we propose a novel chunking technique that preserves the integrity of ornaments at segment boundaries, as detailed below.

Given the dataset $\mathcal{D} = \{(x_i, \{(o_{ij}, f_{ij}, c_{ij})\}_{j=1}^{N_i})\}_{i=1}^{M}$, where x_i denotes the i-th audio file, and each event within x_i represented by its onset o_{ij}, offset f_{ij}, and class label c_{ij}, we propose a chunking strategy for the ornamentation detection task. Here, M is the total number of audio files in the dataset, and N_i is the number of ornaments in the ith file. The objective is to create chunks of length t seconds while ensuring that events are not abruptly truncated at chunk boundaries. Specifically, the chunk duration t is chosen to exceed the maximum duration of any ornament in the dataset,

ensuring that no ornamentation event is truncated even when its onset lies near a segment boundary.

For each audio file x_i, the chunking process begins at time $t_{i0} = 0$ seconds. Each chunk, denoted as s_{ik}, spans the interval $[t_{ik}, t_{ik}+t]$, where t and k denote the duration and index of the chunk corresponding to the ith audio file x_i, respectively. In our proposed chunking method, if an event $e_{ij} = (o_{ij}, f_{ij}, c_{ij})$ starts within the interval $[t_{ik}, t_{ik}+t]$ but ends after $t_{ik}+t$, the next chunk $s_{i(k+1)}$ will start from the onset o_{ij} of this event. Mathematically, if $o_{ij} \leq t_{ik} + t < f_{ij}$, then $t_{i(k+1)} = o_{ij}$.

To ensure that truncated events do not participate in model training, we label the truncated portion as 'don't care' (denoted as †). Note that this 'don't care' label is not part of the dataset annotation; it is introduced only during training to exclude these regions from the loss computation. Specifically, for the k-th chunk s_{ik}, if $t_{ik} \leq o_{ij} < t_{ik} + t$ and $t_{ik} + t < f_{ij}$, then the segment $[o_{ij}, t_{ik} + t]$ within the k-th chunk is labelled as a don't care label, i.e., †. The subsequent chunk $s_{i(k+1)}$ will cover the interval $[o_{ij}, o_{ij} + t]$, and the segment $[o_{ij}, f_{ij}]$ within this chunk will be labelled with the original class label c_{ij}. The proposed chunking technique is illustrated in Figure 3.

Formally, let $\mathcal{S}_i$ be the set of chunks for the audio file x_i, and $\mathcal{L}_i$ be the corresponding labels:

$$\mathcal{S}_i = \{s_{ik}\}_{k=0}^{K_i}, \quad \mathcal{L}_i = \{l_{ik}\}_{k=0}^{K_i}$$

where K_i represents the total count of chunks associated with the audio file x_i, and each chunk s_{ik} is defined as:

$$s_{ik} = [t_{ik}, t_{ik} + t]$$

The labels l_{ik} are assigned as follows:

$$l_{ik} = \begin{cases} c_{ij}, & \text{if } o_{ij} \leq t_{ik} \text{ and } f_{ij} \leq t_{ik} + t \\ \dagger, & \text{if } o_{ij} \leq t_{ik} \text{ and } t_{ik} + t < f_{ij} \end{cases}$$

To further process the audio chunks, we extract chromagram features [43], denoted as $\mathcal{X} = \{X_k\}_{k=1}^{L}$, for all L chunks across the dataset. Here, $L = \sum_{i=1}^{M} K_i$ denotes the total number of chunks across all M audio files. Each $X_k \in \mathbb{R}^{F \times T}$, with F denoting the number of frequency bins and T representing the number of time frames. Let the corresponding labels be $\mathcal{Y} = \{Y_k\}_{k=1}^{L}$, where each $Y_k \in \mathbb{R}^{T}$ encodes the ground truth for each time frame of X_k. The set $(\mathcal{X}, \mathcal{Y})$ is used for training and testing the model.

B. Model

We introduce the Encoder-Decoder Temporal Convolutional Network (ED-TCN) model, inspired by Lea et al. [35] to our task, with chromagram features as input. The authors in [35] designed ED-TCN specifically for the task of action segmentation in video signals. Since our task is similar to action segmentation [44], [45] with respect to the temporal dependency of events, we customize the ED-TCN model to fit our task by introducing periodic padding [46] and dilated convolution inspired by the Wavenet [47] architecture.

The ED-TCN model comprises the following components:

1) Encoder: The encoder consists of L layers denoted by $E^{(l)} \in \mathbb{R}^{F_l \times T_l}$, where F_l is the number of filters applied for the convolution operation in l^{th} layer and T_l is the number of time steps in the corresponding layer. The encoder part of the network applies a series of dilated convolutions to the input chromagram features, $\mathcal{X}$ (Figure S1 in supplemental materials). These convolutions progressively downsample the temporal dimension such that $T_l = \frac{1}{2}T_{l-1}$ using max pooling with pool size of 2 time frames Let $W = \left\{W^{(i)}\right\}_{i=1}^{F_l}$ for $W^{(i)} \in \mathbb{R}^{d \times F_{l-1}}$ where W is the set of convolutional filters and d is the filter size. Let the corresponding bias vector be $b \in \mathbb{R}^{F_l}$. Given the output from the previous layer, $E^{(l-1)}$, the lth layer is computed as

$$E^{(l)} = \sigma(W * E^{(l-1)} + b) \tag{1}$$

where σ is the non-linear activation function and $*$ denotes the convolution operation.

2) Decoder: The architecture of the decoder is similar to that of the encoder. The pooling layer is simply replaced with an upsampling layer. Upsampling is utilized to ensure that the number of time frames in the input layer corresponds to those in the output layer. The decoder also consists of L layers denoted by $D^{(l)} \in \mathbb{R}^{F_l \times T_l}$ where $l \in \{L, \ldots, 1\}$.

3) Dilated Convolutions: To effectively capture long-range temporal dependencies, we incorporate dilated convolutions into the encoder of the ED-TCN model. Unlike standard convolutions that process consecutive time steps, dilated convolutions apply the filter over input features with defined gaps, known as dilation rate, between elements.

4) Periodic Padding: To leverage the periodic nature of musical notes, we introduce a custom padding layer, `PeriodicPadding`, which replaces the standard zero padding used in convolution operations. This periodic padding ensures that the convolution operation respects the cyclic structure of the extracted chromagram features, improving the model's ability to detect repetitive patterns. Mathematically, the periodic padding operation can be described as:

$$PeriodicPadding = \begin{bmatrix} X_{\text{top}} \\ X \\ X_{\text{bottom}} \end{bmatrix}$$

where X is the chromagram feature matrix $\in \mathbb{R}^{F \times T}$, $X_{top} \in \mathbb{R}^{p \times T}$ is the set of last p rows of the X, and $X_{bottom} \in \mathbb{R}^{p \times T}$ is the set of first p rows of the X such that $X_{top} = X[-p:]$ and $X_{bottom} = X[:, p]$ where p is the padding size.

5) Classification Layer: At the end of the decoder, we apply a Time-distributed dense layer with C neurons corresponding to the number of classes. This dense layer acts as the classifier, mapping the final feature map to the class probabilities for each time frame.

The classification layer applies a linear transformation followed by a softmax activation to produce the class probabilities such that:

$$\hat{Y}_t = \text{softmax}(U D_t^{(1)} + b) \tag{2}$$

where $U \in \mathbb{R}^{C \times F_1}$ is the weight matrix, $b \in \mathbb{R}^{C}$ is the bias vector, and $D_t^{(1)}$ represents the last layer of decoder, and

TABLE II
OVERVIEW OF EXPERIMENTAL DATA SPLITS USED TO EVALUATE GENERALIZATION ACROSS SINGERS AND RĀGAS. S_1^{tr} AND S_2^{tr} DENOTE THE TRAINING SUBSETS OF SINGER 1 (S_1) AND SINGER 2 (S_2) DATA, RESPECTIVELY, WHILE S_1^{te} AND S_2^{te} REPRESENT THE CORRESPONDING TESTING SUBSETS.

Exp	Training Data	Testing Data	Testing Type
1	S_1^{tr} + S_2^{tr}	S_1^{te} + S_2^{te}	General
2	S_1^{tr}	S_1^{te}	Intra-Singer
3	S_2^{tr}	S_2^{te}	Intra-Singer
4	S_1	S_2	Inter-Singer
5	S_2	S_1	Inter-Singer
6	S_1, Rāga Bageshree	S_1, Rāga Bhoopali	Rāga-Specific
7	S_1, Rāga Bhoopali	S_1, Rāga Bageshree	Rāga-Specific
8	S_2, Rāga Bhairav	S_2, Rāga Bhoopali	Rāga-Specific
9	S_2, Rāga Bhoopali	S_2, Rāga Bhairav	Rāga-Specific

$\hat{Y}_t \in \mathbb{R}^C$ represents the predicted class probabilities for each time frame $t \in \{1, 2, \ldots, T\}$.

Let the ED-TCN model be denoted by f_θ where θ are the model parameters. The parameters θ are updated using the gradient descent algorithm: $\theta \leftarrow \theta - \alpha \nabla_\theta L(f_\theta)$, where $\alpha \in \mathbb{R}^+$ is the learning rate and L is the categorical cross-entropy loss defined as :

$$L = -\frac{1}{\sum_{t=1}^{T} \mathbb{1}_{\{Y_t \neq \dagger\}}} \sum_{t=1}^{T} \mathbb{1}_{\{Y_t \neq \dagger\}} Y_t \log(\hat{Y}_t) \qquad (3)$$

where $Y_t \in \mathbb{R}^C$ denotes the one-hot representation of the ground truth label for t^{th} time frame of the chromagram X_t and $\hat{Y}_t = f_\theta(X_t)$ denotes the predicted probability distribution over classes as defined in 2. The indicator function $\mathbb{1}_{\{Y_t \neq \dagger\}}$ denotes that while loss calculation, only those time-frames are considered whose labels are not 'don't care' labels †.

VI. EXPERIMENTS

Since the audio files have variable lengths, we segment them into 10-second chunks using the chunking technique discussed in Section V-A, followed by extracting chromagram features. For computing chromagrams, we calculate the Short-time Fourier transform (STFT) of the audio chunks using a 4096-point FFT, a 35 ms sliding Hanning window with a hop-size of 17.5 ms. Since ornamentation detection relies heavily on capturing continuous pitch structures, we examine chromagrams with 12 and 120 bins. Unless specified otherwise, we use 120-bin chromagram features throughout all experiments. We evaluate our model across different train–test splits shown in Table II; details are discussed in Section VII-A.

A. Baseline

We adopt the model proposed in [3] as the baseline, re-implemented using 120-dimensional chromagram features $X \in \mathbb{R}^{F \times T}$ since our dataset consists of monophonic vocal signals where pitch-based representations are more appropriate than mel-spectrograms; all other architectural components, including the (3×3) convolution kernel and (2,1) maxpooling, were kept identical to the original baseline. We use three sets of 2D-Convolution, BatchNormalization, and Maxpooling layers with [32, 64, 128] kernels of size (3, 3). Posterior to it, we apply two bidirectional-GRU layers, each having 8 recurrent units. We use ReLU non-linearity after both convolutional and recurrent layers. Following the bi-GRU layers, we apply a Time-distributed dense layer with softmax activation having $C = 6$ neurons corresponding to six classes. The model is trained for 100 epochs using the Adam optimizer with a learning rate of 0.001.

B. Proposed Model

The input to the proposed ED-TCN model is again chromagram $X \in \mathbb{R}^{F \times T}$ as mentioned above. The encoder consists of L = 4 1D-Convolutional layers having W =[32, 64, 128, 256] number of filters with ReLU activation at each layer. We also apply dilation [47] at each convolutional layer with a linearly increasing dilation rate [1, 2, 3, 4] at each layer. The decoder follows a similar architecture to the encoder. It consists of 1D-Convolutional layers having [256, 128, 64, 32] number of filters with ReLU activation at each layer. The size of each 1D-Convolutional filter is 5. Dilation rates are applied in the opposite sequence to those in the encoder. A Spatial-Dropout layer with a dropout rate of 0.3 is used in both the encoder and decoder. A Time-Distributed dense layer with softmax having C = 6 neurons follows the decoder. The model is trained to 3000 epochs using the Adam optimizer with a learning rate of 0.001.

We test our model across different splits of the dataset, including general overall evaluation, intra-singer, inter-singer, and also *rāga*-specific evaluation, as shown in Table II, detailed in section VII. To further evaluate the generalizability of our model to real-world IAM performances, we fine-tune the best-performing model on the Prasar Bharati dataset [16]. In this setup, we first use DEMUCS [48] to extract the vocal stems from the polyphonic audio recordings. We then use our model, originally trained on the ROD dataset, as a fixed feature extractor and fine-tune only the final layers on the new, unseen dataset.

C. Evaluation Metrics

We report Precision, Recall, and F1-score averaged over the classes for our task. Since it is practically infeasible to annotate the boundaries of events precisely, the model struggles to predict the exact boundary of the events. It is a common practice to use a collar around the event boundaries, to address this issue. The authors in [9], [38] use a 200 ms collar around the event boundaries during evaluation. We adopt similar collar-based evaluation metrics for our task.

VII. RESULTS AND DISCUSSIONS

A. Evaluation Across Data Split Scenarios

To comprehensively evaluate our model's generalization capability, we conduct experiments under multiple data-split scenarios, as detailed in Table II.

1) General Evaluation: In Experiment 1, we consider a merged dataset from both singers, with a random 70/20/10 train-test-validation split. As shown in row-1 of table III, we observe a significant improvement of about 16% with collar and 13% without collar over the baseline, highlighting the effectiveness of our proposed TCN model.

TABLE III
PERFORMANCE COMPARISON OF THE BASELINE AND PROPOSED MODEL ACROSS NINE EVALUATION EXPERIMENTS FOR THE ROD DATASET, REPORTED WITH AND WITHOUT A 200 MS COLLAR. IAS DENOTES INTRA-SINGER, IS DENOTES INTER-SINGER, AND RS DENOTES RĀGA-SPECIFIC. ALL VALUES ARE REPORTED IN PERCENTAGE (%)

Experiment	Without Collar						With 200 ms Collar					
	Baseline			Proposed Model			Baseline			Proposed Model		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1: General	74.92	73.78	74.35	**90.56**	**89.56**	**90.06**	81.08	75.28	78.71	**94.14**	**89.20**	**91.98**
2: IAS	65.91	64.86	65.38	**90.15**	**89.04**	**89.59**	71.95	66.83	69.30	**93.54**	**89.40**	**91.42**
3: IAS	69.77	68.90	69.33	**92.87**	**92.25**	**92.55**	76.39	70.80	73.49	**95.04**	**92.41**	**93.71**
4: IS	64.57	63.72	64.14	**82.77**	**82.05**	**82.41**	70.97	65.88	68.33	**86.52**	**82.70**	**84.57**
5: IS	52.16	51.48	51.82	**80.98**	**79.91**	**80.44**	59.45	54.74	57.00	**87.87**	**81.19**	**84.44**
6: RS	70.40	68.89	69.64	**75.41**	**74.39**	**74.90**	81.44	71.93	76.39	**82.32**	**76.02**	**79.05**
7: RS	66.71	65.66	66.18	**79.00**	**78.18**	**78.59**	73.78	67.90	70.71	**84.32**	**79.27**	**81.72**
8: RS	58.93	58.34	58.63	**81.95**	**81.38**	**81.66**	63.63	60.19	61.86	**85.59**	**82.03**	**83.77**
9: RS	57.95	57.13	57.54	**77.08**	**76.37**	**76.72**	64.22	59.62	61.83	**81.95**	**77.45**	**79.64**

2) Intra-Singer Evaluation: In Experiments 2 and 3, we train and evaluate the model separately on each singer's recordings using distinct train-test splits. Rows 2 and 3 in Table III show very similar or somewhat better performance as compared to Experiment-1. This indicates that despite the added variability from combining singer styles in Experiment-1, the model generalizes well, unlike the baseline which struggles under the same conditions.

3) Inter-Singer Evaluation: In Experiments 4 and 5, we assess the model's generalization across singers by training on one singer's data and testing on the other's. As expected, the baseline performance drops significantly in this cross-singer setup. However, our TCN-based model maintains relatively strong performance, indicating its ability to learn underlying structural patterns of ornamentations that transfer across stylistic differences.

4) Rāgā-Specific Evaluation: In Experiments 6 to 9, we evaluate the model's robustness across different *rāgās* by training on one raga and testing on another for the same singer. Given that ornamentation styles vary significantly across ragas, some performance drop is expected. As shown in Table III, while the baseline model drops to around 60% F1-score, our proposed model consistently maintains strong performance around 80%, demonstrating better generalization to unseen *rāgā* contexts.

Figure 5 presents the confusion matrix for our best-performing configuration, which utilizes F = 120, 'don't care' labelling, periodic padding, and dilated convolutions. The confusion matrix reveals patterns consistent with theoretical expectations and annotation trends. For instance, *Kaṇ* is often misclassified as *Mīnd*, which is expected since *Mīnd* is essentially a slower, more extended version of *Kaṇ* articulated with a glide. Similarly, *Andolan* is frequently confused with *Mīnd*, as both share a gliding character, with *Andolan* essentially being a repeated or oscillatory variant of a *Mīnd*. As shown in Figure 8, if we isolate any one glide segment within the *Andolan*, it closely resembles what is labelled a *Mīnd* in our dataset. Furthermore, the model experiences significant confusion between *Murkī* and *Kaṇ*, since *Murkī* can be conceptualized as a sequence of repeated *Kaṇ*. Additionally, the flexible nature of *Murkī*, by definition, complicates the classification, causing the

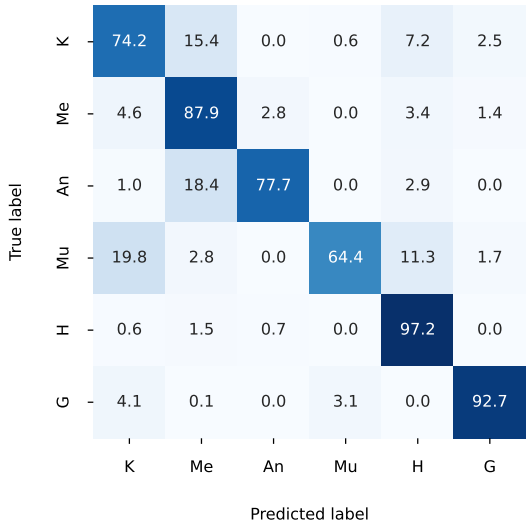


Fig. 5. Confusion matrix illustrating the performance of the proposed model on the ROD dataset. The rows represent the true labels (ground truths), and the columns represent the predicted labels. K, Me, An, Mu, H, and G represent *Kaṇ*, *Mīnd*, *Andolan*, *Murki*, *Nyās Svar*, and *Gamak* respectively. Each cell show the percentage (%) of true-class frames (row) classified as the corresponding predicted class (column)

model to perform least effectively on this class. In contrast, the model accurately classifies *Nyās Svar* and *Gamak*, due to their clearly defined structural characteristics and extended temporal duration. The confusion matrix for baseline model is shown in Supplemental material (Figure S2).

B. Evaluation on concert recordings

To evaluate the model's generalizability to real-world Indian classical music performances, we fine-tuned our best-performing model on a set of 52 audio recordings obtained from Prasar Bharati archives. We compare it with a baseline model trained from scratch on this dataset, as well as with a model pre-trained on the ROD dataset and directly tested on the new dataset without any fine-tuning. Despite the huge domain shift, the model demonstrates satisfactory performance after minimal adaptation, as summarized in Table IV.

TABLE IV
PERFORMANCE COMPARISON OF THE BASELINE AND PROPOSED MODELS, REPORTED WITH AND WITHOUT A 200 MS COLLAR FOR THE PRASAR BHARATI DATASET. FT DENOTES FINE-TUNING. ALL VALUES ARE IN PERCENTAGE (%)

Model	Without Collar			With 200 ms Collar		
	P	R	F1	P	R	F1
Baseline	53.94	53.41	53.68	60.83	56.39	58.52
Proposed Model (without FT)	61.32	60.75	61.03	65.02	61.98	63.47
Proposed Model (with FT)	**65.26**	**64.90**	**65.80**	**69.06**	**66.18**	**67.59**

TABLE V
ABLATION STUDY ON PROPOSED MODEL PERFORMANCE FOR EXPERIMENT 1 (†, PP, DI-CONV, AND F DENOTE DON'T CARE LABELLING, PERIODIC PADDING, DILATED CONVOLUTION, AND CHROMA BINS RESPECTIVELY). F=120, UNLESS SPECIFIED OTHERWISE. ALL METRIC VALUES ARE REPORTED IN PERCENTAGE (%)

Configuration	P	R	F1
Baseline	74.92	73.78	74.35
Without †	79.55	78.50	79.02
Without PP	87.82	86.36	86.82
Without Di-Conv	87.61	86.68	87.14
With †, PP, and Di-Conv, F = 12	87.28	86.33	86.80
With †, PP, and Di-Conv	**90.56**	**89.56**	**90.06**

C. Ablation studies

Table V presents the results of ablation studies conducted for Experiment-1 evaluated without a collar. The first row represents baseline CRNN model, while all others represent our proposed ED-TCN model. The baseline CRNN model yields an F1-score of 74.35, which is reasonable given the complexity of the ornamentation detection task. However, all configurations of our proposed model outperform the baseline significantly. Among these, we observe that increasing the chromagram resolution from the standard 12 bins to 120 further improves performance. As shown in Figure 7, a sample for ornament *Andolan* is more clearly represented with 120 bins, capturing the fine-grained, continuous pitch variations more effectively. This improvement is also reflected in the F1-score increase from 86.80 to 90.06. While ornamentations are expressed as pitch fluctuations, pitch contours are often unreliable due to octave and voicing errors in pitch estimation. The chromagram, by compressing pitch information within one octave and implicitly encoding the tonic, offers a more robust and musically meaningful representation. Empirically, the chromagram-based model outperformed the pitch-contour-based model on our dataset, as shown in Supplemental Material (Figure S5).

We now investigate the impact of different components in our method to identify optimal conditions for our proposed model. We study the effectiveness of 'don't care labelling', the number of chroma bins, periodic padding, and dilated convolution in our proposed TCN-based model by systematically dropping one component at a time. The Rows 2, 3, and 4 in Table V show the usage of all other components except the one mentioned there. It is assumed that F = 120 unless mentioned otherwise.

1) 'Don't care' labelling: When using both periodic padding and dilated convolutions, but without the 'don't care' labelling strategy, the model still outperforms the baseline by approximately 4%, indicating an immediate benefit of the temporal modelling framework. However, adding 'don't care' labelling provides an increased performance of 11%, which is a huge jump. In our dataset, we observe that if we do not incorporate 'don't care' labeling for chunking, we end up with 31% of the chunks containing at least one fragmented or partially labelled event. This means that the model would receive a number of misleading examples and would learn from them. By employing overlapping chunks and 'don't care' labels, we maintain the structural integrity of audio events and exclude ambiguous regions from training, thereby facilitating more effective model training and enhancing overall performance, as is reflected in Table V. It is also notable that 'don't care' labelling consistently boosts F1 scores by 7–8%, more than either dilated convolutions or periodic padding, highlighting its effectiveness for such tasks. We further examine the impact of 'don't-care' labeling on each class in our dataset. As observed, long-duration ornaments are minimally affected by abrupt truncation and, therefore, show limited improvement with the proposed *"don't care labeling"* strategy. In contrast, short-duration ornaments such as *Kaṇsvar*, *Mīnd*, and *Murkī* are highly susceptible to truncation, which often distorts their temporal structure and makes them difficult to recognize. Consequently, these ornaments benefit significantly from the *"don't care labeling"* strategy, as reflected in their performance improvement shown in Figure 6. Instances illustrating the effect of 'don't care' labeling are shown in Supplemental Material (Figure S6, S7).

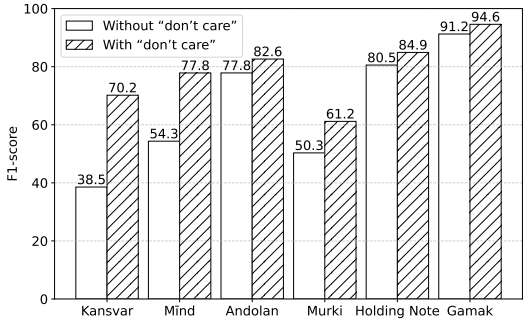


Fig. 6. Effect of the "don't care labeling" strategy on class-wise detection performance (F1-score (reported in %)).

2) Periodic Padding: To investigate the impact of padding strategies on our model's performance, we conduct experiments comparing periodic padding with conventional zero padding. Our results indicate a decline of more than 3% when using zero padding as compared to periodic padding as shown in the Table V. This degradation can be attributed to the intrinsic periodic structure of musical notes. By preserving this periodicity through periodic padding, the extracted chromagram features more accurately reflect the underlying musical patterns, thereby enhancing the model's ability to learn and generalize from the data.

3) Dilated Convolutions: In this scenario, we evaluate the effect of dilated convolutions compared with standard convolutions. We find the performance to decrease by 3% while

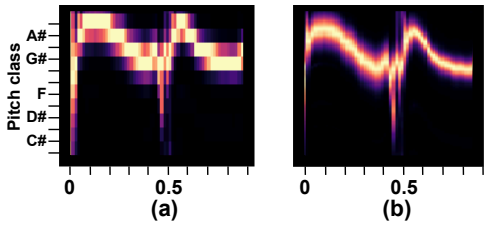


Fig. 7. Comparison of (a) 12-dimensional and (b) 120-dimensional chromagrams for the same audio excerpt taken from ROD dataset.

using normal convolutions. This decrease is due to the coarser scale offered by dilated convolutions, enabling the model to capture a broader context and more intricate patterns in the data. Instances illustrating the effect of 'don't care' labeling are shown in Supplemental Material (Figure S8, S9).

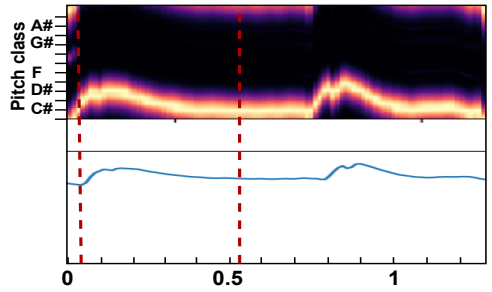


Fig. 8. The figure depicts an instance of *Andolan* with a cycle (marked in red lines) resembling *Mīnd*. The x-axis represents time (in sec).

D. Subjective Analysis

This section outlines the key sources of error in the predictions of our model and highlights the underlying complexities in the dataset and the task as follows:

- **Subjectivity in singing style across singers and *Rāga*:** A major challenge in the task of automatic ornamentation detection in IAM arises from the inherent subjectivity in singing styles, both across different singers and across different *rāgas*. Ornaments like *Mīnd* or *Andolan* can be executed with varying degrees of pitch deviation, duration, and subtlety, depending on the artist's stylistic preference or the aesthetic norms of a specific *rāga*. For instance, in the ROD dataset, we observed that in *rāga Bhairav*, the *Andolan* around the notes *komal re* and *komal dha* is typically wide and slow, while in *rāga Bhūpāli*, glides between the notes *ga* and *pa* are often very subtle and shorter.
- **Inherent similarity in structure of ornaments:** Many ornamentations have similar structures inherently. For example, repeating *Mīnd* more than two times constitutes *Andolan*. Moreover, increasing the tempo of *Andolan* constructs *Gamak*. One such instance is depicted in Figure 8. In this figure, the two cycles of *Andolan* take on a shape similar to that of two consecutive *Mīnd*. This confuses the model to predict the cycles of *Andolan* as *Mīnd* individually and vice versa. This intricacy is evident in Figure 5. The red lines in Figure 8 indicate the resemblance of *Mīnd* with the cycles of *Andolan*.

VIII. Conclusion and Future Scope

In this work, we introduce the task of automatic recognition of vocal ornamentations in Hindustani Classical Music (HCM). We present the ROD dataset containing expert-annotated HCM recordings labelled with six distinct ornamentation types and the associated *rāga* names. Using this dataset, we develop an Encoder-Decoder Temporal Convolutional Network (ED-TCN) model that outperforms a CRNN baseline and generalizes effectively across different singers. Our work establishes a benchmark for automatic ornamentation detection in HCM and highlights the contribution of each model component through ablation studies. We observe that the model benefits from periodic padding, fine-grained chromagrams, and dilated convolutions, with 'don't care' labelling offering the strongest advantage by addressing temporal fragmentation.

The proposed ornamentation detection framework has broad applications across music pedagogy, expressive singing voice synthesis, singer identification, and content-based music recommendation. The detected ornament labels may also serve as control signals for expressive singing voice synthesis, enabling style-conditioned vocal generation. Future work can focus on exploring more robust domain adaptation strategies to improve model performance on real-world performance audio. Expanding datasets would enhance generalizability, but the annotation process remains challenging due to its high cost, intensive effort, and reliance on expert knowledge. The subjective nature of ornament interpretation and structural overlaps among ornament types also demand hybrid approaches that combine musicological insight with data-driven learning for better performance.

IX. Acknowledgements

We appreciate Kajal and Renu Chavan for their roles as expert vocalists and annotators. We also thank Parampreet, Sparsh, and Sharah for annotation support and Shivnarayan for managing the dataset creation. This work was supported by Prasar Bharati, India's public broadcasting agency.

References

[1] S. Giraldo and R. Ramírez-Melendez, "A machine learning approach to ornamentation modeling and synthesis in jazz guitar," in *Machine Learning and Music Generation*. Routledge, 2018, pp. 21–40.
[2] K. Banerjee, A. Patranabis, R. Sengupta, and D. Ghosh, "Ornamentation in hindustani vocal music," in *FRSM*, 2023, pp. 125–135.
[3] Y. Yamamoto, J. Nam, and H. Terasawa, "Analysis and detection of singing techniques in repertoires of j-pop solo singers," in *ISMIR*, 2022.
[4] ——, "Primadnn: A characteristics-aware dnn customization for singing technique detection," in *EUSIPCO*. IEEE, 2023, pp. 406–410.
[5] H. Purwins, "Analysis and classification of ornaments in north indian (hindustani) classical music," *Master's thesis, UPF, Barcelona*, 2010.
[6] Y. Yamamoto, J. Nam, and H. Terasawa, "Deformable cnn and imbalance-aware feature learning for singing technique classification," *arXiv preprint arXiv:2206.12230*, 2022.
[7] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM TASLPRO*, vol. 25, no. 6, pp. 1291–1303, 2017.
[8] C.-C. Kao, W. Wang, M. Sun, and C. Wang, "R-crnn: Region-based convolutional recurrent neural network for audio event detection," *arXiv preprint arXiv:1808.06627*, 2018.
[9] S. Kumar, S. Jaiswal, P. Singh, and V. Arora, "Automatic detection and analysis of singing mistakes for music pedagogy," *Authorea Preprints, TechRxiv:10.36227/techrxiv.23269502.v2*, 2025.

[10] J. F. Merchán Sánchez-Jara, S. González Gutiérrez, J. Cruz Rodríguez, and B. Syroyid Syroyid, "Artificial intelligence-assisted music education: A critical synthesis of challenges and opportunities," *Education Sciences*, vol. 14, no. 11, p. 1171, 2024.

[11] S. Holland, "Artificial intelligence in music education: A critical review," *Readings in music and artificial intelligence*, pp. 239–274, 2013.

[12] J. Zhao, L. Q. H. Chetwin, and Y. Wang, "Sintechsvs: A singing technique controllable singing voice synthesis system," *IEEE/ACM TASLPRO*, vol. 32, pp. 2641–2653, 2024.

[13] Y.-J. Luo, C.-C. Hsu, K. Agres, and D. Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," in *IEEE ICASSP 2020*, pp. 3277–3281.

[14] N. Shikarpur, K. M. Dendukuri, Y. Wu, A. Caillon, and C.-Z. A. Huang, "Hierarchical generative modeling of melodic vocal contours in hindustani classical music," *arXiv preprint arXiv:2408.12658*, 2024.

[15] D. Y. Loni and S. Subbaraman, "Robust singer identification of indian playback singers," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, pp. 1–14, 2019.

[16] P. Singh and V. Arora, "Explainable deep learning analysis for raga identification in indian art music," *IEEE Transactions on Audio, Speech and Language Processing (TASLPRO)*, vol. 33, pp. 2302–2311, 2025.

[17] J. Bor, S. Rao, W. V. D. Meer, and J. Harvey, "The raga guide: a survey of 74 hindustani ragas," *Nimbus Records with Rotterdam Conservatory of Music*, 1999.

[18] C. Gupta and P. Rao, "Objective assessment of ornamentation in indian classical singing," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2011, pp. 1–25.

[19] P. Proutskova, C. Rhodes, T. Crawford, and G. Wiggins, "Breathy, resonant, pressed–automatic detection of phonation mode from audio recordings of singing," *JNMR*, vol. 42, no. 2, pp. 171–186, 2013.

[20] Y. Wang, W. Wei, and Y. Wang, "Phonation mode detection in singing: A singer adapted model," in *IEEE ICASSP 2023*, pp. 1–5.

[21] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset." in *ISMIR*, 2018, pp. 468–474.

[22] K. L. Kim, J. Lee, S. Kum, C. L. Park, and J. Nam, "Semantic tagging of singing voices in popular music recordings," *IEEE/ACM TALPRO*, vol. 28, pp. 1656–1668, 2020.

[23] Y. Xu, W. Wang, H. Cui, M. Xu, and M. Li, "Paralinguistic singing attribute recognition using supervised machine learning for describing the classical tenor solo singing voice in vocal pedagogy," *EURASIP Journal on Audio, Speech, and Music Processing*, no. 1, p. 8, 2022.

[24] V. Kalbag and A. Lerch, "Scream detection in heavy metal music," *arXiv preprint arXiv:2205.05580*, 2022.

[25] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE workshop*, 2017.

[26] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[27] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE MLSP*, 2015, pp. 1–6.

[28] ——, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015.

[29] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[30] S. Kumar, B. Anshuman, L. Rüttimann, R. H. Hahnloser, and V. Arora, "Balanced deep cca for bird vocalization detection," in *IEEE ICASSP*, 2023, pp. 1–5.

[31] P. Chordia, "Segmentation and recognition of tabla strokes." in *conference of ISMIR*, 2005, pp. 107–114.

[32] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, "A review of automatic drum transcription," *IEEE/ACM TASLPRO*, vol. 26, no. 9, pp. 1457–1483, 2018.

[33] G. Ding, F. Sener, and A. Yao, "Temporal action segmentation: An analysis of modern techniques," *IEEE TPAMI*, vol. 46, no. 2, 2023.

[34] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *European conference on computer vision*. Springer, 2016, pp. 47–54.

[35] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 156–165.

[36] S. S. Miryala, K. Bali, R. Bhagwan, and M. Choudhury, "Automatically identifying vocal expressions for music transcription." in *ISMIR*, 2013, pp. 239–244.

[37] The Audacity Team, "Audacity: Free audio editor and recorder," 2024. [Online]. Available: https://www.audacityteam.org/

[38] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[39] J. Harrison, "The gharana system: Lineage in indian classical music," 2023, accessed: 2025-02-22. [Online]. Available: https://www.darbar.org/article/the-gharana-system-lineage-in-indian-classical-music

[40] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM TASLPRO*, vol. 30, pp. 829–852, 2021.

[41] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE ICASSP*, 2015, pp. 5206–5210.

[42] G. E. Poliner and D. P. Ellis, "A classification approach to melody transcription," in *ISMIR*, 2005.

[43] M. Müller, *Fundamentals of music processing: Using Python and Jupyter notebooks*. Springer, 2021, vol. 2.

[44] Y. A. Farha and J. Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2019, pp. 3575–3584.

[45] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3192–3199.

[46] S. Schubert, P. Neubert, J. Pöschmann, and P. Protzel, "Circular convolutional neural networks for panoramic images and laser data," in *IEEE intelligent vehicles symposium (IV)*, 2019, pp. 653–660.

[47] A. van den Oord, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[48] A. Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.

Supplemental Material

Annotation Tool

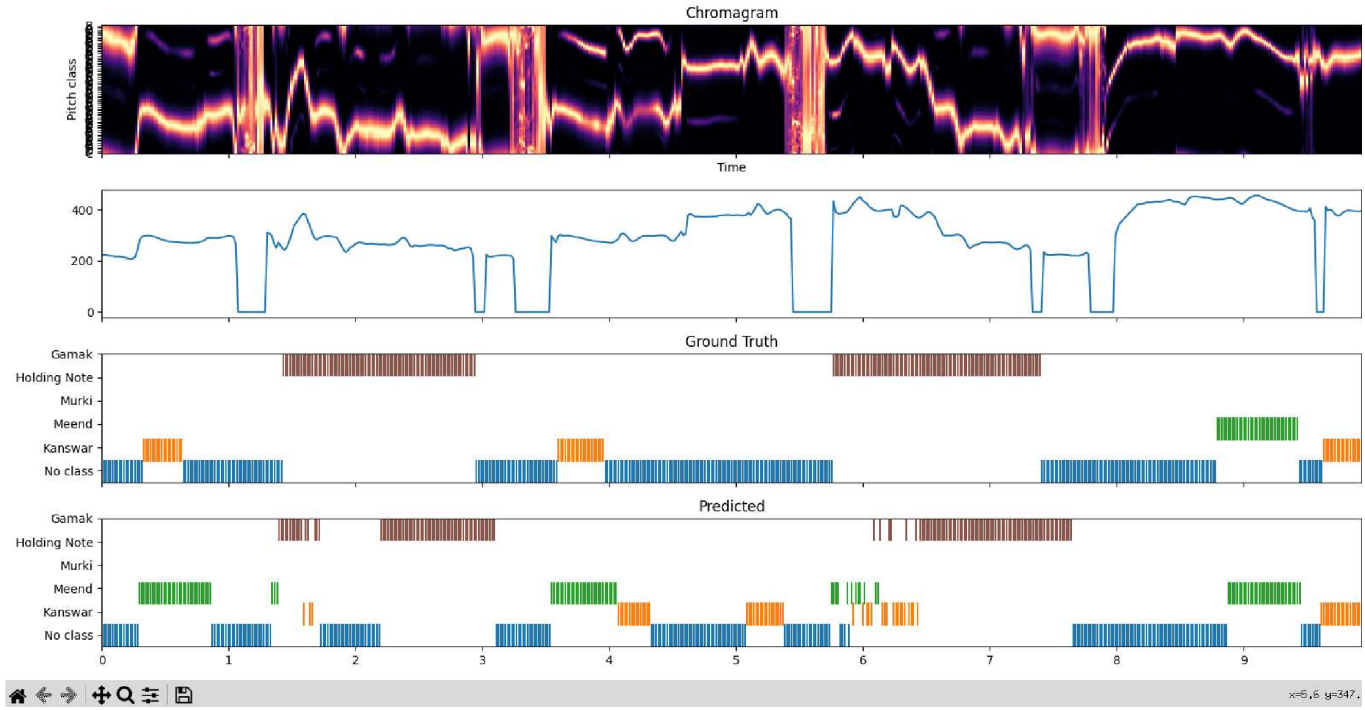
Figure 1 - Figure 4

Figure 1: Custom-built Active-annotation Tool interface



To use the tool, execute the `interface.sh` script. This will launch the active annotation interface window as shown.

Figure 2: Chromagram, pitch contour, ground truth labels, and model predictions

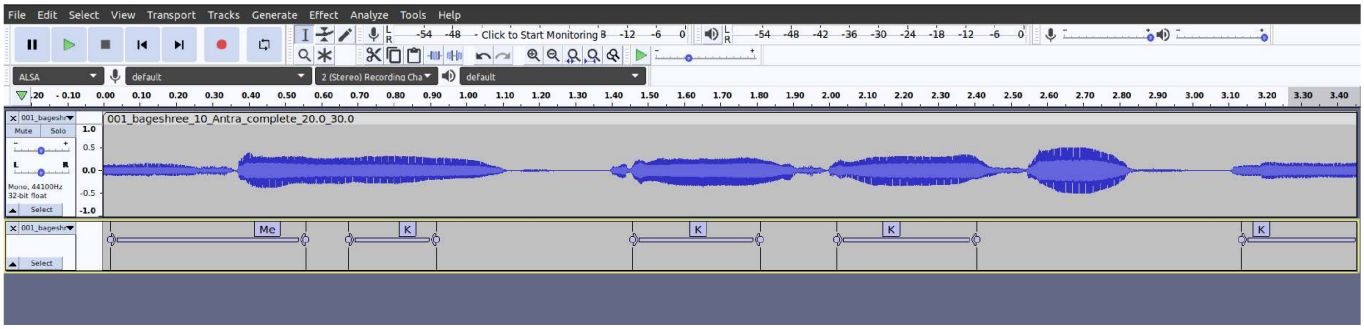


To generate this visualization, the user must:

1. Load the pre-trained model, chromagrams, and audio files.
2. Click the **Predict** button to compute predictions.
3. Enter the index of the audio file to correct.
4. Click the **Plot Results** button.

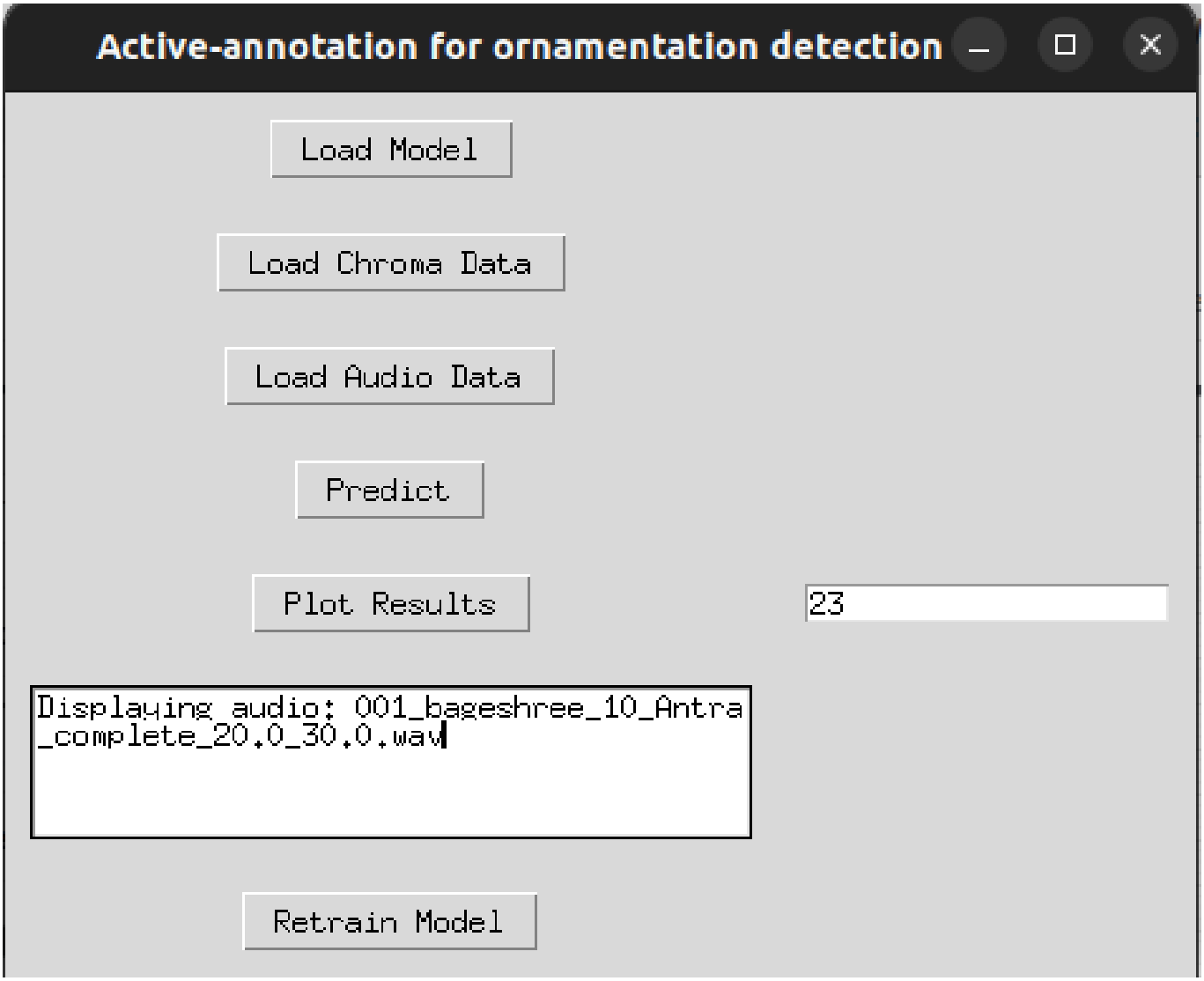
This opens a window displaying the analysis and simultaneously launches **Audacity** for annotation correction.

Figure 3: Audacity annotation interface



Modify the labels using Audacity and export the corrected label files.

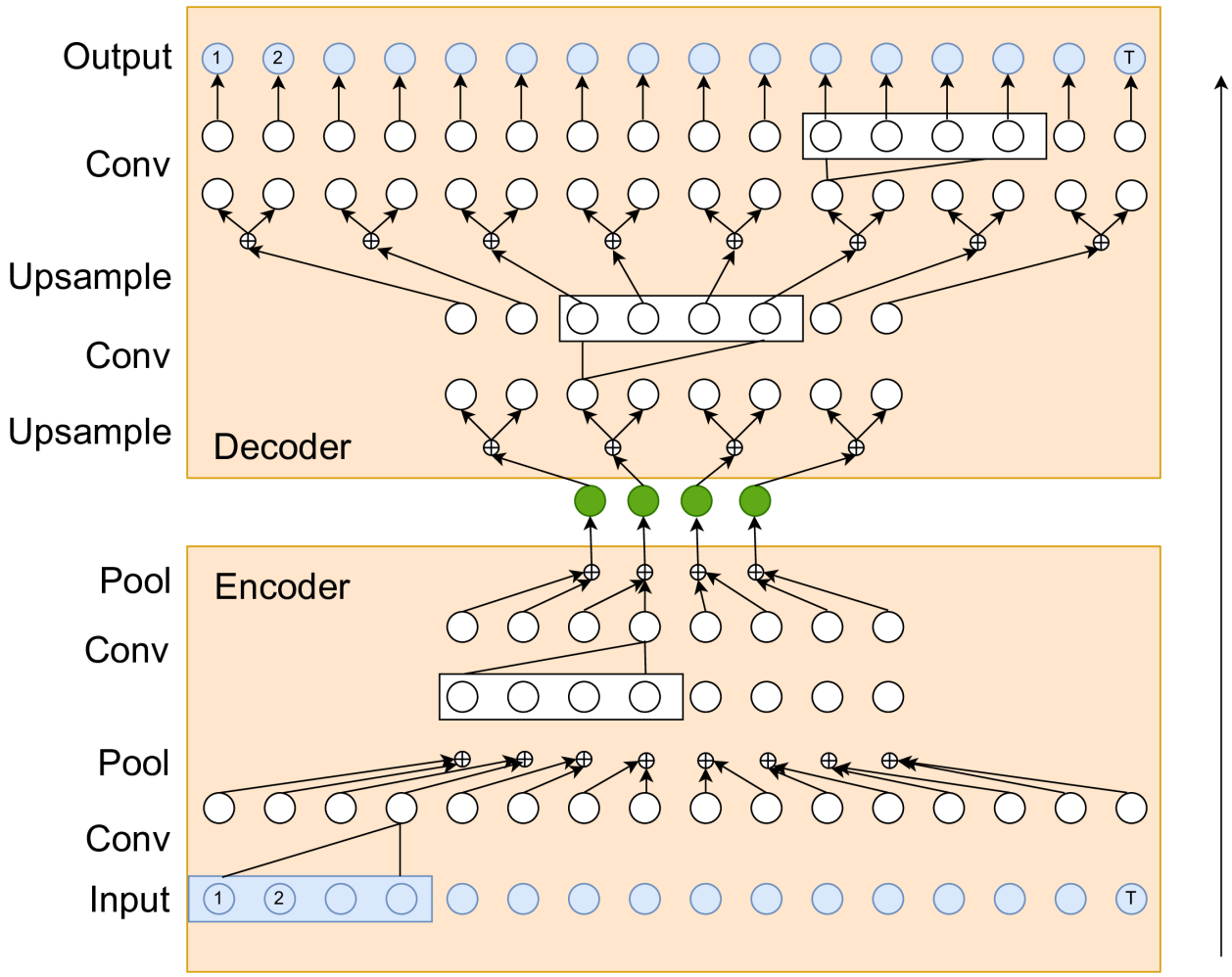
Figure 4: Retrain Model interface



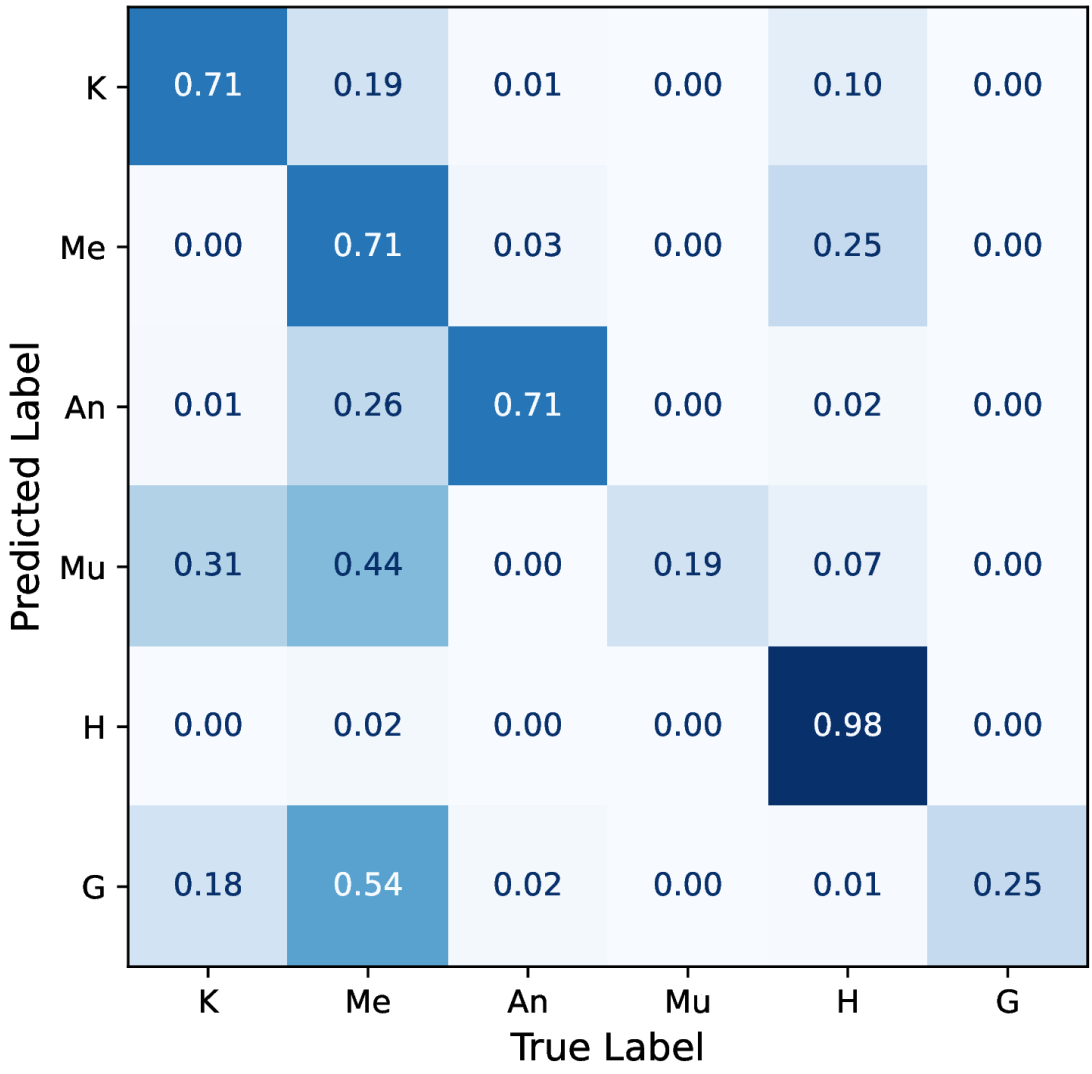
Click the **Retrain Model** button to update the model with the revised labels.

Other Supplimental Material

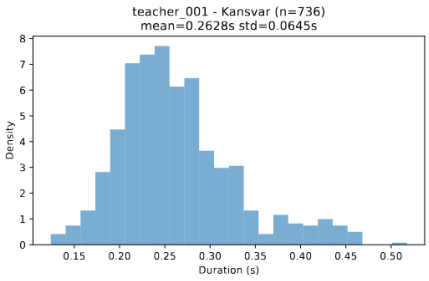
S1: ED-TCN model block Diagram



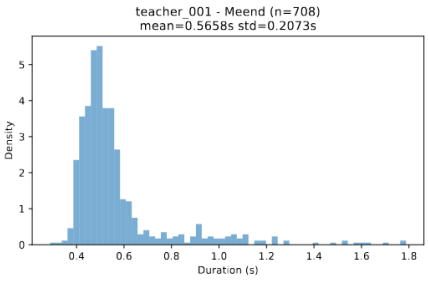
S2: Confusion matrix for baseline



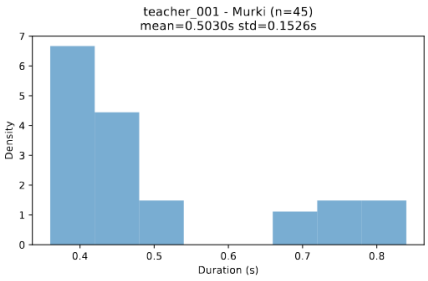
S3: Teacher 1 Ornaments Distribution



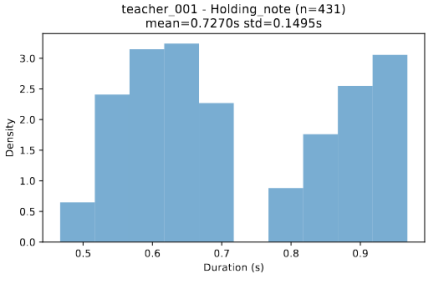
(a) Kansvar



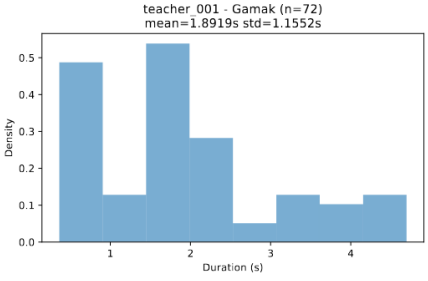
(b) M̄ind



(c) Murki

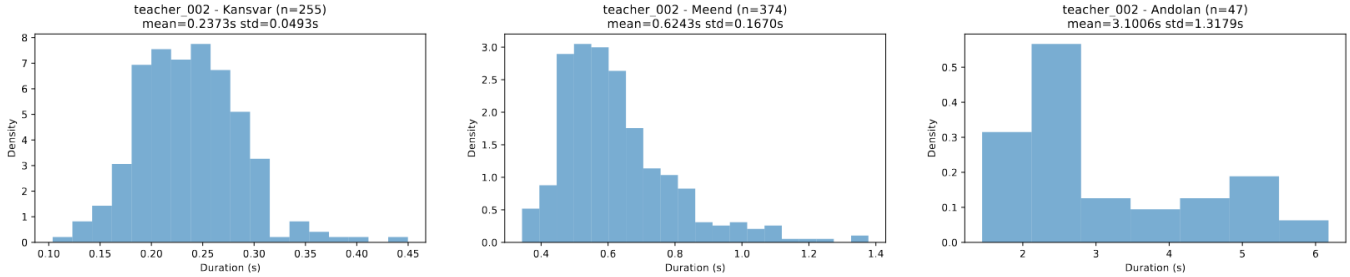


(d) Holding note



(e) Gamak

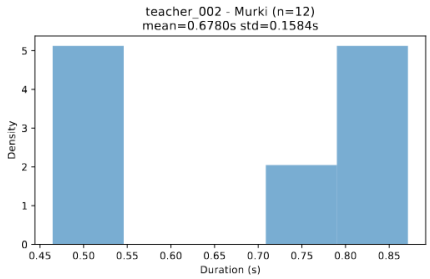
S4: Teacher 2 Ornaments Distribution



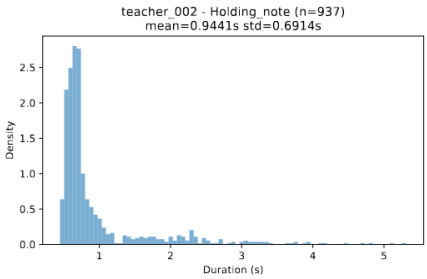
(a) Kansvar

(b) Mīnd

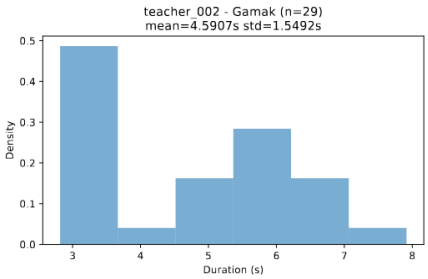
(c) Andolan



(d) Murki



(e) Holding note



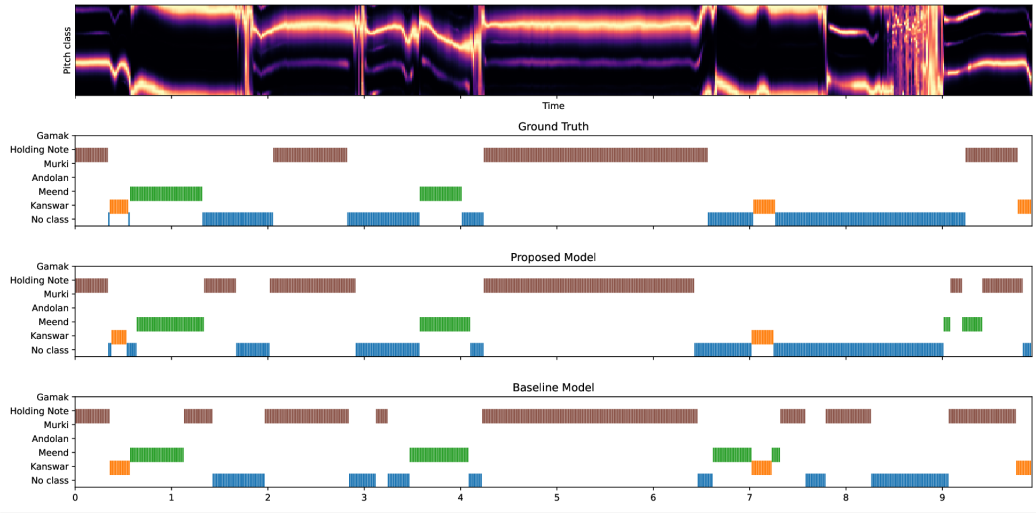
(f) Gamak

S5: Pitch Contour vs. Chromagram Features

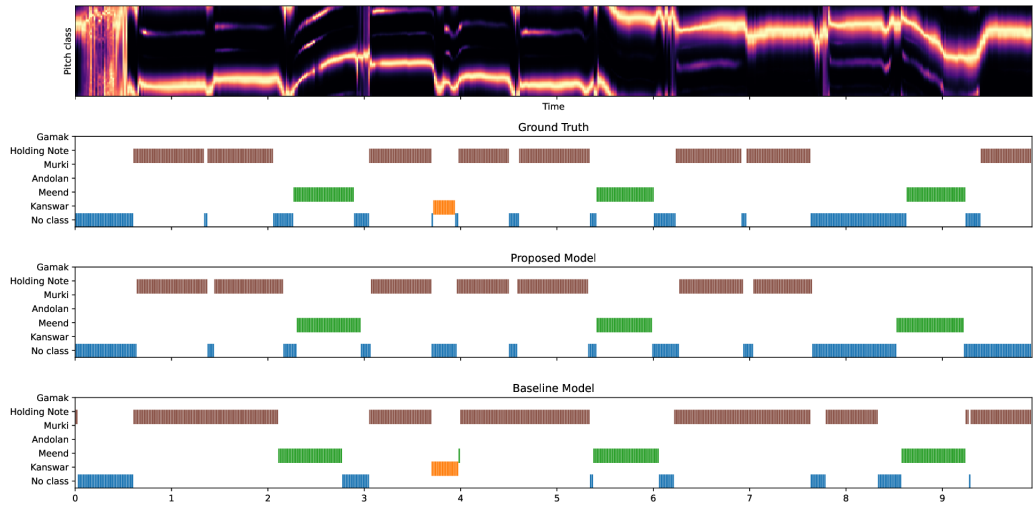
Table: Class-wise F1-scores obtained using pitch contour and chromagram features.

Classes	Pitch Contour	Chromagram
Kansvar	63.87	70.17
Mīnḍ	46.21	77.83
Andolan	16.99	82.62
Murki	37.28	61.16
Holding Note	69.91	84.90
Gamak	84.64	94.58

S6: Illustration of the baseline model predicting abruptly truncated Kansvar and Holding Note. This is handled by proposed model using don't care labelling.



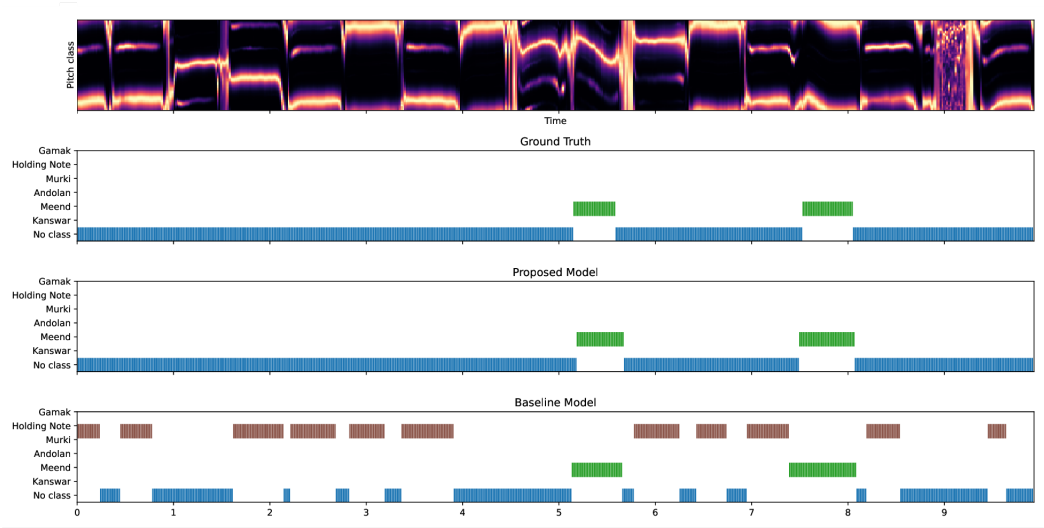
(a) Illustration of the baseline model predicting abruptly truncated *Kansvar* predictions



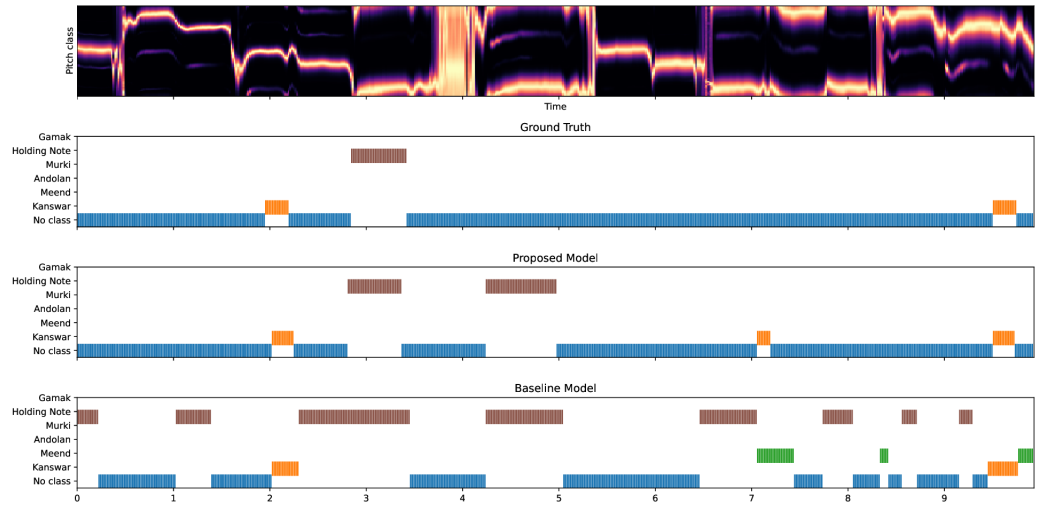
(b) Illustration of the baseline model predicting abruptly truncated *Kansvar* predictions

Figure 5: Effect of *don't-care* labeling. The proposed model learns to predict complete events instead of the abruptly truncated ones predicted by the baseline, improving robustness in real-world applications.

S7: Comparison of false positives. The baseline model produces several spurious detections, while the proposed model significantly reduces them.



(a) Baseline model predicting Holding Notes as False Positive.



(b) Baseline model predicting Holding Notes and M̄ind as False Positive.

Figure 3: Comparison of false positives. The baseline model produces several spurious detections, while the proposed model significantly reduces them.

S8: Comparison of event boundary modeling. The proposed model maintains temporal separation between closely spaced ornaments, unlike the baseline model, which merges them into a single prediction.

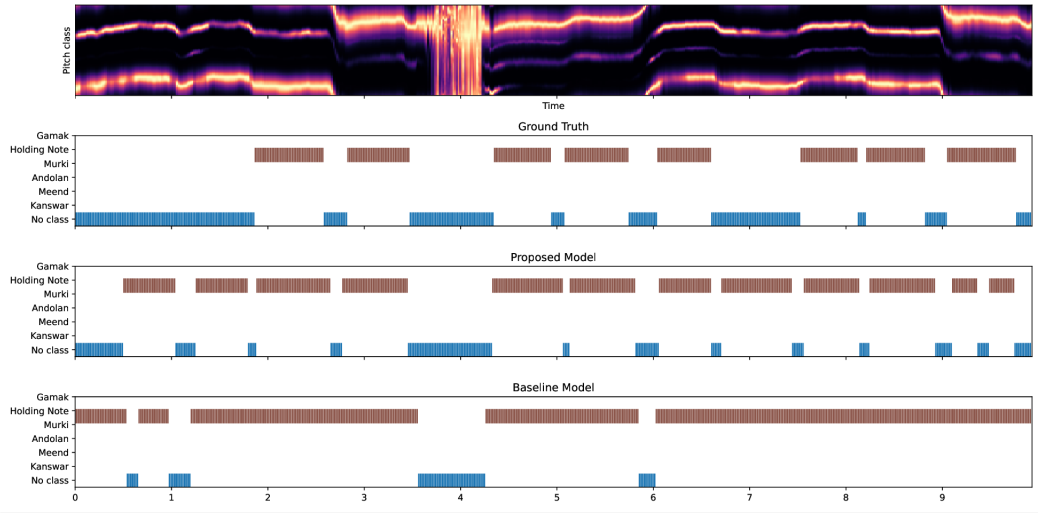
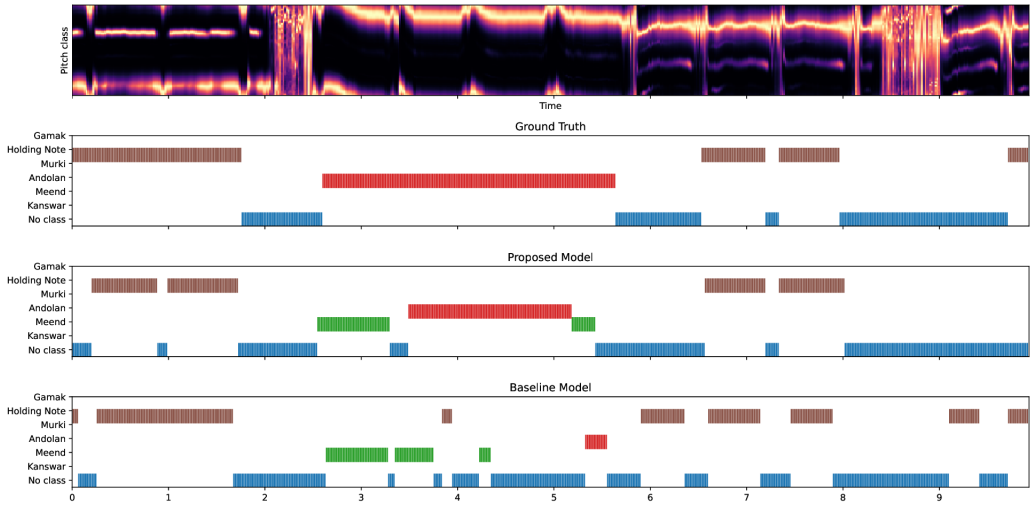
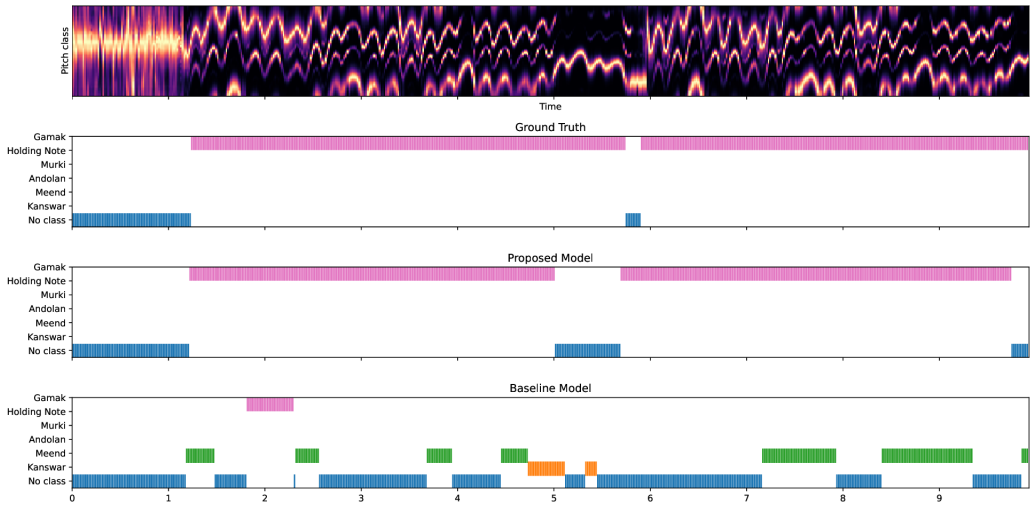


Figure: Comparison of event boundary modeling. The proposed model maintains temporal separation between closely spaced ornaments, unlike the baseline model, which merges them into a single prediction.

S9: Comparison of temporal dependency modeling. The baseline model fails to capture long-range dependencies, while the proposed model maintains temporal coherence across ornaments.



(a) Illustrating the difficulty of the baseline model in detecting long-duration ornamentation: *Andolan*.



(b) Illustrating the difficulty of the baseline model in detecting long-duration ornamentation: *Gamak*.

Figure 4: Comparison of temporal dependency modeling. The baseline model fails to capture long-range dependencies, while the proposed model maintains temporal coherence across ornaments.