Supplementary Details

1. Dataset details

In addition to the dataset [1], we recorded alap data across 8 new singers and use the newly augmented dataset in this work. The data is recorded with 3 cameras however this study includes only analysis of front camera. The recording setup used for the new data is given below

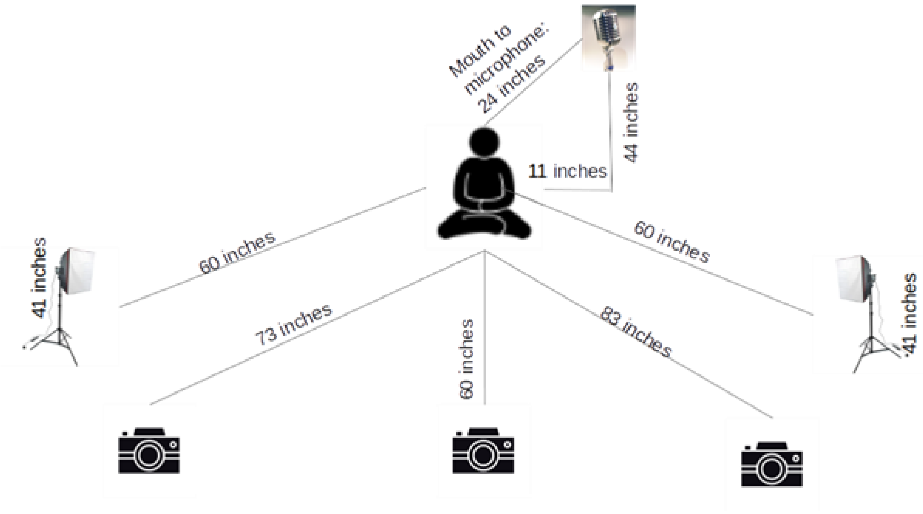

Fig 1: A representation of the setup (not drawn to scale). The singer is in a sitting position on the ground in a cross-legged fashion. There are 3 cameras, two lights and one microphone.

Audio Metadata

1. File Type : Wave (.wav)
2. Sample Rate : 48.000 kHz
3. Bit Depth : 24 bit
4. Audio Codec : AAC

Microphone Model - NEUMANN KMS 105

Video Metadata

1. Format : MP4
2. Resolution : 1920 x 1080
3. Frame Rate : 24 fps
4. Video Codec : H.264 High L4.0

Camera Model : SONY NXCAM HXR-NX5R

Singer and Raga information

The following are the details of the singers (added in this work) using their initials.

Male Singers:

AK / MG / MP / NM

Female Singers:

AP / RV / SS / SM

All of the singers are **right handed** (self declared). The singers in the previous dataset [1] were AG, SCh (female) and CC (male).

Each singer sang 2 Alaps and 1 Pakad for each of the following ragas. Some of the ragas are abbreviated as in parenthesis. The ragas are the same as in [1].

Bageshri (Bag)
Bahar
Bilaskhani Todi (Bilas)
Jaunpuri (Jaun)
Kedar
Marwa
Miyan ki Malhar (MM)
Nand
Shri

2. Obtaining Gesture kinematic variables from video

Our processing is similar to the methodology published in [7]. We extract 11 upper body keypoints via Openpose [5] from each frame of the video. We thus convert keypoint position coordinates for a joint into a time series. We exclude keypoint position samples extracted with a confidence of less than 0.3 and fill in the gaps by linearly interpolating using neighbouring frames. We use a Savitzky-Golay filter [6] of 4th order with 13 point smoothing window to remove jitter. We normalize the position time series for every joint based on the mean and standard deviation for that joint [z-normalization].

Additionally, in this work, we compute the velocities and acceleration profiles of the joints of interest. A smoothened derivative is computed on the 2d wrist position time series using convolution with a biphasic filter as below.

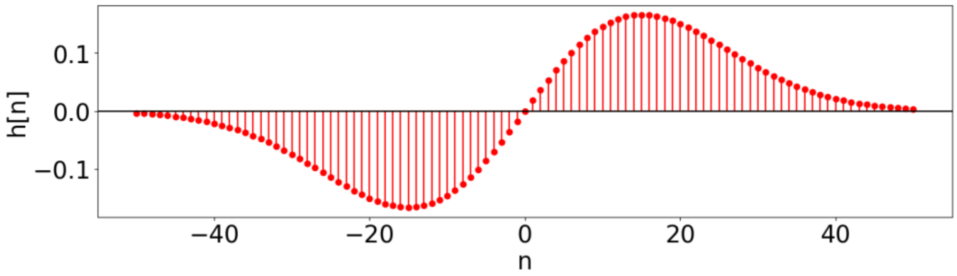

Fig 2 : Plot of impulse response a 101 point bi-phasic filter for velocity and acceleration $\tau_1 = \tau_2$=15 and $d_1 = d_2 = 2$. The sampling rate here is 100 Hz.

The biphasic filter [2,3] is defined by

$$h[n] = \frac{1}{\tau_1\sqrt{2\pi}} e^{-\frac{(n-d_1)^2}{\tau_1^2}} - \frac{1}{\tau_2\sqrt{2\pi}} e^{-\frac{(n-d_2)^2}{2\tau_2^2}}$$

We convolve the biphasic filter for differentiation of the position coordinates along each axis (x,y) for each joint to obtain the velocity for that joint. We take the Euclidean norm of the individual components of velocity and refer to $v[n]$ as velocity for the joint in the rest of the work.

$$v_x[n] = h[n] * \; p_x[n]$$

$$v_y[n] = h[n] * \; p_y[n]$$

$$v[n] = \sqrt{\left(v_x^2[n] + v_y^2[n]\right)}$$

where $p_x[n]$ and $p_y[n]$ are the co-ordinates of a joint along x and y axes respectively and "*" indicates the convolution operation.

We use the same biphasic filter for differentiation of the velocity along each axis to obtain the acceleration along that axis. We take the Euclidean norm of the individual components of acceleration and refer to $a[n]$ as acceleration for the joint in the rest of the work.

$$a_x[n] = h[n] * v_x[n]$$

$$a_y[n] = h[n] * \; v_y[n]$$

$$a[n] = \sqrt{\left(a_x^2[n] + a_y^2[n]\right)}$$

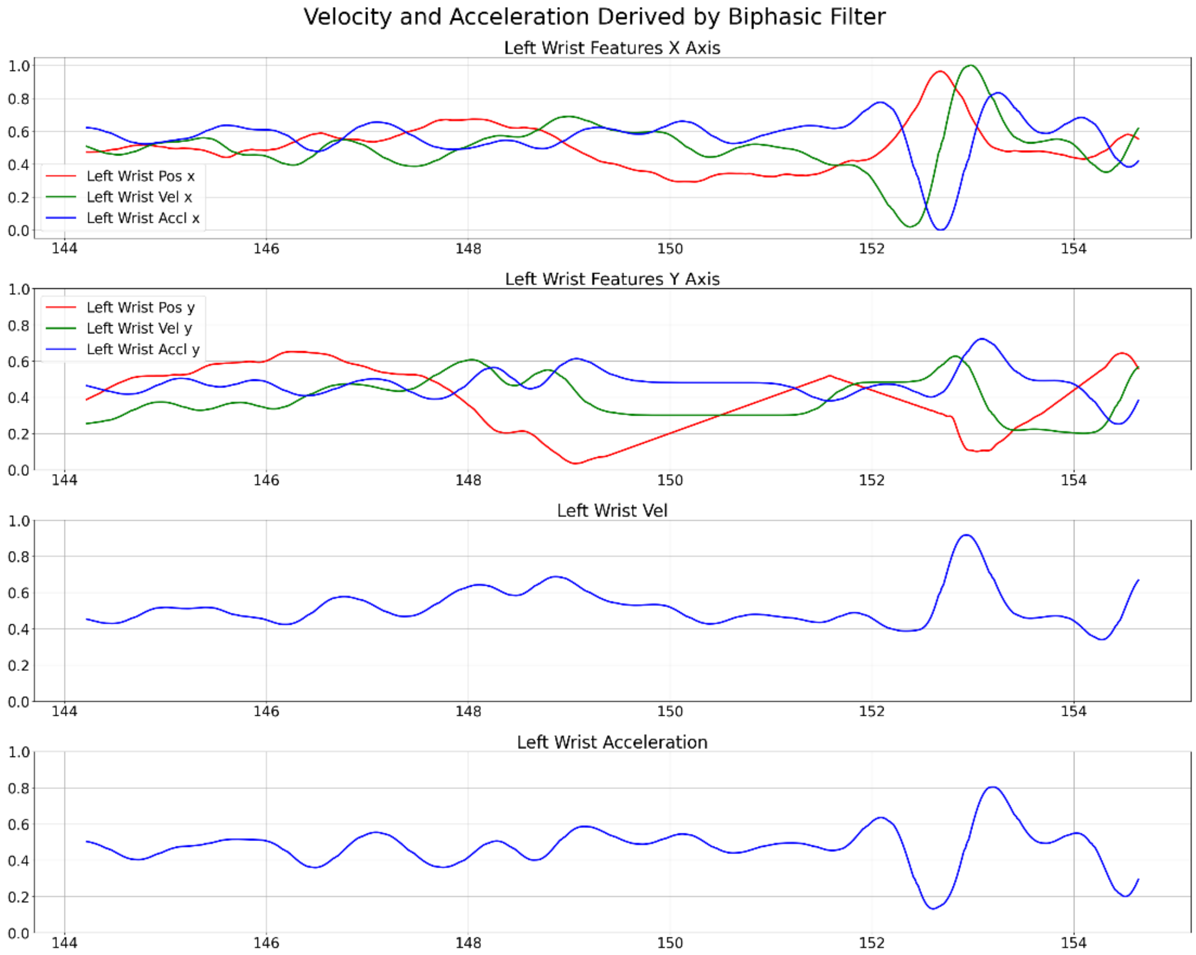

Fig 3: Kinematic features for left wrist for a particular SDS. X axis is in time (s) of the original alap of which this is a part. **First plot:** X-components of position, velocity, acceleration for the left wrist. **Second plot:** Y-components of position, velocity, acceleration for the left wrist. **Third plot**: Magnitude of velocity (speed) of the left wrist computed as the Euclidean norm of the vector velocity having both x and y components. **Fourth plot:** Magnitude of acceleration of the left wrist computed as the Euclidean norm of the vector acceleration having both x and y components. An identical processing is done for the right wrist.

3. Hyperparameter Search Range for Stable Note Classifier

Hyperparameter	Range
No. Of Conv1D layers (including BN+ReLU)	[1,2,3,4]
No. Of filters in each layer	[2,6,10,14]
Filter size for Conv1D layers	[3,5,7]
No. Of Dense Layers after Conv1D	[0,1]
No of nodes in Dense layer	[2,6,10,14]

Table 3.1: Hyperparameter Search Range for different stable Note models.

Convolution layers consist of a Batch Normalization and a ReLU also. There is a Global Average Max pooling layer after the Convolutional layers. There is a final output dense layer with sigmoid activation for getting the probability of stable note.

Training loss is binary cross entropy and Keras Tuner is used for best model identification based on F1 Score using Bayesian Optimization with 25 trials. Optimizer is Adam with default parameters[1] and batch size is 32.

Hyperparameter	P	P+V	V+A	P+V+A
Number of Conv layers	1	2	4	3
Num Filters in Conv Layer 1	2	14	10	2
Filter Size in Conv Layer 1	5	7	7	5
Num Filters in Conv Layer 2	-	6	2	6
Filter Size in Conv Layer 2	-	5	5	5
Num Filters in Conv Layer 3	-	-	6	14
Filter Size in Conv Layer 3	-	-	5	10
Num Filters in Conv Layer 4		-	2	-
Filter Size in Conv Layer 4		-	3	-
No. Of Dense Layers after Conv1D	0	1	0	-
No. Of node in dense layer	-	2	-	

Table 3.2 Chosen best hyperparameters for various feature combinations

4. Supplementary Tables for Stable Note Classification

Feature	Duration	Precision.	Recall	F1
P	>1s	28.4	73.9	**41.0**
	>2s	21.0	44.4	**28.5**
P+V	>1s	30.9	42.6	35.8
	>2s	32.3	30.9	**31.6**
V+A	>1s	30.1	62.2	**40.6**
	>2s	30.6	37.5	**33.7**
P+V+A	>1s	29.4	65.3	**40.5**
	>2s	29.8	45.1	**35.9**

Table 3S.1 Precision / Recall /F1-Score (%) for best classification model for various combinations of kinematic features of position(P), velocity(V) and acceleration(A) . The total number of segments is 33484 (23.1 % stable notes) and there are 12620 which are longer than 2s duration (14.7% stable). Bold font indicates the model is significantly better than a random baseline (p<0.001). The random baseline F1 score for durations >1s is 37.6\% and for >2s is 25.6\%.

[1] Adam (keras.io)

SInger	Count	Stable %	Precision	Recall	F1
AG	788	10.2	19.4	35.0	25.0
AK	908	22.9	32.5	31.2	31.9
AP	820	17.6	22.6	24.3	23.4
CC	1628	9.1	19.9	38.5	26.3
MG	876	42.9	48.3	45.2	46.7
MP	884	52.9	64.7	61.1	62.9
NM	1168	1.4	2.1	18.8	3.8
RV	1600	8.0	17.1	43.0	24.5
SCh	1888	1.3	3.9	45.8	7.1
SM	828	27.1	39.6	50.9	44.5
SS	1232	3.2	6.9	32.5	11.4

Table 3S.2 F1-score / Precision / Recall (%) - SInger-wise classification details for the best model (P+V+A) as identified in Table 3 in main paper are shown here. All durations are > 2s.

Comments:- The singers (e.g. NM / SS) have very low proportion of stable notes in training data and so any singer specific gesturing is unlikely to be captured by the model.

Raga	Count	Stable %	Precision	Recall	F1
Bag	1408	11.6	21.9	43.9	29.2
Bahar	1392	4.6	9.6	32.8	14.9
Bilas	1452	20.4	39.6	45.6	42.4
Jaun	1336	17.1	32.6	46.9	38.5
Kedar	1188	13.8	28.8	45.1	35.2
MM	1580	14.4	27.9	42.1	33.6
Marwa	1660	12.8	26.4	50.0	34.5
Nand	1316	15.2	36.7	49.0	42.0
Shri	1288	23.3	40.4	42.7	41.5

Table 3S.3 F1-score / Precision / Recall (%) - Raga-wise classification details for the best model (P+V+A) as identified in Table 3 in main paper are shown here. All durations are > 2s.

Comments:- We would expect Bahar to score lower (tendency for faster, livelier movements) and Shri higher (serious raga with both Re and Pa tending to be held quite stable for significant durations). This is also indicated in the corresponding proportion of steady notes for these two ragas.

Type	Count	Correctly Predicted	Recall
Tonic	592	202	34.1
Non-Tonic	1264	635	50.2
Tonic + Non Tonic	1856	837	45.1

Table 3S.4 Recall (%) for stable notes splits between tonic and non-tonic considering segments >2s. Details for the best model (P+V+A) as identified in Table 3 in main paper are shown here

Comments:- We think that the lower Sa in particular could be accompanied by nondescript gestures (as nothing interesting is happening), whereas other notes are being gesturally 'fixed' and therefore are easier to predict. In that case, we would perhaps expect upper Sa to be more like the non-tonic level; but the model may not have enough data to do this classification better.

5. Supplementary Tables for Raga Phrase Classification

	gmD - DTW_LR (2).			**r/P - DTW_IND (8)**			**P\R - DTW_IND (8)**		
Singer	**Like**	**Total**	**Acc**	**Like**	**Total**	**Acc**	**Like**	**Total**	**Acc**
AG	112	165	55.8	91	185	50.8	124	217	40.1
AP	93	171	50.3	79	196	59.7	69	201	65.2
CC	58	147	56.5	94	228	58.8	59	163	66.3
MG	71	140	45.7	74	160	53.8	36	180	73.9
MP	75	145	54.5	81	195	58.5	50	188	73.9
NM	67	148	39.9	85	220	61.4	47	221	76.9
RV	88	179	58.1	130	271	52	96	205	55.6
SCh	87	144	52.1	90	211	57.3	100	271	64.6
SM	113	179	50.3	128	184	30.4	82	181	58.6
SS	89	192	55.2	88	203	56.7	82	201	57.2
Overall	**944**	**1771**	**52.4**	**1035**	**2303**	**56.1**	**817**	**2157**	**65.2**

Table 4S.1 : Count of “Like” phrases, total count and accuracy (%) per singer for gesture based phrase classification. Details for only the best model as identified in Table 4 in the main paper are shown here.

Comments:- The distribution is relatively uniform across singers except the case of SM for r/P, and AG in P\R. More discussion and examples of the singer AG for P\R are presented with the accompanying videos.

1. References:-

[1] M. Clayton, J. Li, A. R. Clarke, M. Weinzierl,L. Leante, and S. Tarsitani, “Hindustani raga and singer classification using pose estimation,” 2021. [Online]. Available: https://doi.org/10.17605/OSF.IO/T5BWA

[2] D. J. Hermes, “Vowel-onset detection,” The Journal of the Acoustical Society of America, vol. 87, no. 2, pp. 866–873, 1990.

[3] P. Rao, T. P. Vinutha, and M. A. Rohit, “Structural segmentation of alap in dhrupad vocal concerts,” Transactions of the International Society for Music Information Retrieval, vol. 3, no. 1, 2020.

[4] T. O’Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi et al., “Kerastuner,” https://github.com/keras-team/keras-tuner, 2019

[5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “Openpose: Realtime multi-person 2d pose estimation using part affinity fields,” IEEE Transactions on Pattern Analysis and Machine Intelligence,vol. 43, no. 1, pp. 172–186, 2021.

[6] Press, William H., and Saul A. Teukolsky. "Savitzky-Golay smoothing filters." *Computers in Physics* 4.6 (1990): 669-672.

[7] Clarke, A., Weinzierl, M., & Li, J. (2021). Pose estimation for Raga (Version v1.0.1) [Computer software]. https://github.com/DurhamARC/raga-pose-estimation