FOUR-WAY CLASSIFICATION OF TABLA STROKES WITH MODELS ADAPTED FROM AUTOMATIC DRUM TRANSCRIPTION

Rohit M A **Amitrajit Bhattacharjee** **Preeti Rao**
Department of Electrical Engineering
Indian Institute of Technology Bombay, India
{rohitma, amitrajit, prao}@ee.iitb.ac.in

ABSTRACT

Motivated by musicological applications of the four-way categorization of tabla strokes, we consider automatic classification methods that are potentially robust to instrument differences. We present a new, diverse tabla dataset suitably annotated for the task. The acoustic correspondence between the tabla stroke categories and the common popular Western drum types motivates us to adapt models and methods from automatic drum transcription. We start by exploring the use of transfer learning on a state-of-the-art pre-trained multiclass CNN drums model. This is compared with 1-way models trained separately for each tabla stroke class. We find that the 1-way models provide the best mean f-score while the drums pre-trained and tabla-adapted 3-way models generalize better for the most scarce target class. To improve model robustness further, we investigate both drums and tabla-specific data augmentation strategies.

1. INTRODUCTION

Tabla, a ubiquitous part of the North Indian art music ensemble, comprises two drums that can be struck singly or together with a variety of articulations to give rise to sequences of individual and compound strokes of changing timbre, termed *bols*. With a set of between 10-20 distinct tabla bols (depending on playing style) found in practice, the bols have been traditionally viewed as single entities of different timbres, and tabla transcription addressed as a monophonic timbre recognition problem [1].

The earliest work on tabla transcription was reported by Gillet et. al. [2] who modelled stroke spectra by a 4-mixture Gaussian Mixture Model for 10-category classification using Hidden Markov Models (HMM). Chordia [3] extended this work by targeting a larger, more diverse dataset, and using neural network (NN) and tree-based classifiers to categorize strokes based on spectral and temporal envelope features. Both works mention the difficulty of generalizing across instruments, and report lower scores on tabla sets not seen in training. Later work [4] used frame-level mel-frequency cepstral coefficients (MFCC) to capture bol timbre in an HMM model in a classification task on a single tabla set. More recent works that make use of NN and tree-based bol classifiers [5–8] are restricted either in their use of small datasets, or the absence of any instrument-independent performance evaluation.

An important taxonomic level for tabla sounds is based on which of the two drums is struck and the manner of striking, giving rise to the three classes: resonant treble (right drum), resonant bass (left drum), and damped (either drum); the right & left are with respect to a right-handed player. That is, the specific manner gives rise to either a damped stroke with a sharp and short-duration sound or a pitched (resonant) stroke with ringing sound, which can further be pitch modulated in the case of the left drum. The different timbres of the tabla bols are obtained by individual or combinations of basic strokes, with the combination of resonant bass and treble (resonant both) being especially important. In the archetypal drum pattern known as the *theka*, subsections of the rhythmic cycle are chiefly discriminated by the presence or absence of right and left drum resonant strokes [9,10]. The associated classification has been useful in the empirical analyses of tabla accompaniment in khyal vocal performances [11]. Motivated by the musicological applications of the above categorization of tabla sounds, a 4-way stroke classification task was previously defined exploiting the acoustic characteristics of the strokes [12]. A training dataset of labelled tabla solo recordings was created to train a random forest classifier

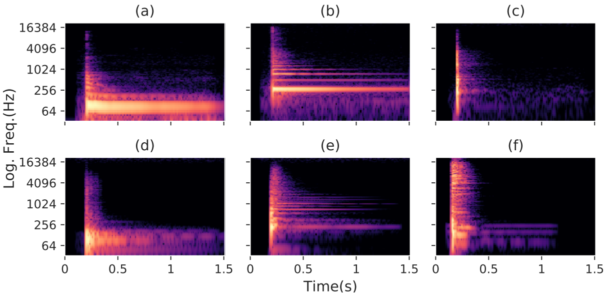

Figure 1: Spectrograms of samples (at f_s=44.1 kHz) of the 3 basic tabla strokes(top) and drum types(bottom). Note the similarity between (a) Resonant Bass & (d) BD, (b) Resonant Treble & (e) SD, and (c) Damped & (f) HH

Attribution: M. A. Rohit, A. Bhattacharjee, and P. Rao, "Four-way Classification of Tabla Strokes with Models Adapted from Automatic Drum Transcription", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

Category	Bols	Bass	Treble	Drumkit
D	Ti-Ta, Te-Re, Tak, Ke, Tra, Kda	D/Nil	D/Nil	HH
RT	Na, Tin, Tun, Din	D/Nil	R	SD
RB	Ghe, Dhe, Dhi, Dhet	R	D/Nil	BD
B	Dha, Dhin	R	R	BD+SD

Table 1: Tabla stroke categories, corresponding bols, constituent stroke types (Resonant/Damped/none) on each tabla drum, and western drum equivalents. D - damped, RT - resonant treble, RB - resonant bass, B - resonant both.

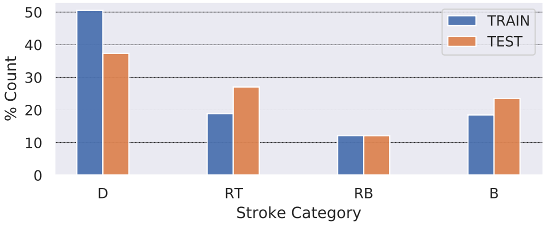

Figure 2: Distribution of stroke categories in our dataset.

with a large set of stroke specific acoustic features, which was then evaluated on a tabla accompaniment test set.

In this work, we recognise the similarity of the reduced category tabla stroke classification problem to the automatic drums transcription (ADT) task with its considerable published work focused on transcribing the 3 main percussion instruments in Western popular music – bass (BD), snare (SD), and hi-hat (HH) [13, 14]. Figure 1 illustrates the correspondence of these drums with the bass, treble, and damped tabla strokes, respectively. Starting with segment-and-classify approaches based on extracting suitable acoustic features for classification from automatically segmented drum tracks using onset detection, more recent methods for ADT adapt deep-learning based onset detection models, trained to directly predict the instrument along with its onset location [14]. We extend previous available work on 4-way tabla stroke category detection to use recently proposed convolutional neural network (CNN) models comprising the state-of-the-art in ADT.

Our chief new contributions include significantly expanding the available training dataset of tabla solo recordings with new instruments, and investigating CNN architectures from ADT literature for the tabla stroke classification task. In an attempt to alleviate training data scarcity, we explore domain adaptation or transfer learning with an available pre-trained multiclass CNN drums model [15]. To counter target class imbalance, we also investigate architecture optimizations with a bank of single-stroke (binary) CNN classifiers [16]. Finally, we explore a number of data augmentation approaches including new tabla-specific transformations inspired by drum-specific augmentation methods from ADT [17]. We present next the dataset used in this work, followed by the classification and data augmentation methods and, finally, the results.

	Source	# tablas	Duration	# strokes
Train	Train-set of [12]	3	18 min.	6,680
	Suppl. data of [18]	3	16 min.	5,178
	New	4	42 min	14,742
	Total	**10**	**76 min.**	**26,600**
	Test-set of [12]	3	20 min.	4,470

Table 2: The various subsets in the train and test datasets.

2. DATASET

An important application of the present work is in classifying tabla strokes played in accompaniment to lead music. We thus use an existing dataset of realistic tabla accompaniment recordings to test our methods. While we would prefer matched training data, concert audios are not readily available in bleed-free multi-track format. And creating such a dataset is challenging not only to record, but also to annotate due to the lack of a precise score. Therefore, we resort to the use of tabla solo playing and build upon previous datasets to create a diverse training set. Table 2 lists the sources of our train and test datasets with a common sampling rate of 16 kHz. The target classes used in this work, common bols that they map to, and the types of strokes played simultaneously on each drum to realise them appear in Table 1. For D, we have damped strokes on either one or both drums. RT and RB strokes produce resonant sounds on the corresponding drum and may be accompanied by a damped stroke on the other drum.

Testing: The test set consists of 10 pieces of only the tabla accompaniment recorded in perfect isolation to pre-recorded solo Hindustani vocal tracks. It contains 20 minutes of audio and nearly 4,500 strokes. These recordings, made on 3 unique tabla sets by 2 different artists, are diverse in terms of tuning, *tala* (metre), and tempo.

Training and Cross-Validation (CV): Solo compositions and common theka patterns recorded from 7 different tabla-sets are added to the training dataset from [12]. Out of these, 3 are from a previous study [18] for which written scores are available but are not time-aligned with the audios. The 4 others were newly recorded for this work by different artists. In order to achieve better diversity, we choose instruments of sufficiently different tuning, include a variety of playing styles, and cover a wide tempo range. Annotation was carried out by automatically aligning the composition score (supplied by artists) with the audios, and replacing the bols with corresponding target stroke categories (Table 1). Given the imperfect score-stroke matching [3,4], labels were manually verified to assign the same category to similar sounding bols. The dataset spans a total audio duration of about 1.25 hours and contains 26,600 strokes. To perform hyperparameter tuning, we split our training dataset into 3 nearly equal-sized disjoint folds, with all recordings from a single tabla set assigned to a single fold, providing instrument independent validation. The folds are similar in the distribution of stroke categories, tonic, and tempo.

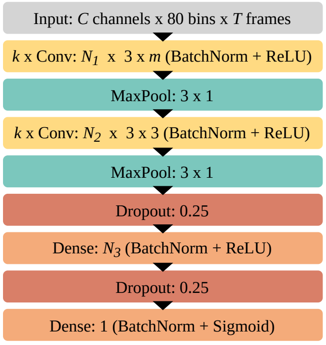
Input: *C* channels x 80 bins x *T* frames
k x Conv: N_1 x 3 x *m* (BatchNorm + ReLU)
MaxPool: 3 x 1
k x Conv: N_2 x 3 x 3 (BatchNorm + ReLU)
MaxPool: 3 x 1
Dropout: 0.25
Dense: N_3 (BatchNorm + ReLU)
Dropout: 0.25
Dense: 1 (BatchNorm + Sigmoid)

Figure 3: General CNN architecture for 1-way model tuning experiments (N_2=2xN_1). *k* is # repetitions of the layer.

Variant	Hyperparameter values
Baseline	C=3, T=15 (150 ms), k=1, m=7, N_1=16, N_3=128
↑context	T=21 (210 ms)
Mid-channel	C=1 (middle)
↑conv filters	N_1=32
↑dense units	N_3=256
↑conv filt. + ↑dense units	N_1=32, N_3=256
2x conv layers	k=2, m=3

Table 3: The various hyperparameter settings in the tuning experiments of the 1-way CNN model (of Figure 3).

Figure 2 shows the distribution of strokes across the four target categories in the train and test sets. We observe a significant imbalance in both, with damped strokes being the most numerous and resonant bass being the least. Although the distributions are similar across datasets, differences in the playing styles (solo vs accompaniment) are likely to contribute to some train-test mismatch.

3. METHODS

We now present the CNN based classification models, their input-output representations, and the training and hyperparameter tuning experiments. Subsequently, we outline the different augmentation methods devised.

3.1 Classification Models

Two CNN-based approaches from drums transcription are compared - a 3-way model [19], and a bank of separate 1-way models for each target class [16]. In the former, we experiment with fine-tuning available pre-trained models as well as training new models with the same architecture from scratch. With the 1-way approach, a model for each stroke category is trained from scratch and their hyperparameters are optimised separately.

3.1.1 3-way Classification

We use the four 3-way CNN models from the python library *madmom* [20], each of which is trained on a different subset of the MIREX17 drums dataset [15]. During training of the 3-way CNN, the fourth 'resonant both' (B) label in tabla is replaced by simultaneous onsets in RB and RT (see Table 1). Model outputs are post-processed during evaluation to obtain 4-class predictions, by replacing RB and RT onsets predicted within 10 ms with B.

Based on the common assumed roles for a CNN's conv and dense layers of feature extractor and classifier respectively [21, 22], we explore two transfer learning strategies to fine-tune (FT) the pre-trained (PT) models on our smaller (by ≈3x) dataset: (a) FT all dense layers while keeping all conv layers frozen at PT values, and (b) FT all layers. Under (b), we study three approaches - uniform, differential, and disjoint. In uniform and differential FT, all layers are simultaneously tuned, with the learning rate (*lr*) kept same for all layers in the former, and different for conv and dense layers in the latter. Disjoint FT refers to the alternating (rather than simultaneous) tuning until convergence of the dense and conv layers, in order to reasonably constrain the updatable parameters at any time. Other fine-tuning combinations with tuning only a subset of dense layers were not found to be favorable. While fine-tuning all layers uniformly has been previously used in audio event tagging [23], the differential and disjoint approaches are motivated from speech recognition [24, 25]. Finally, we consider also the PT initialisation of dense layers in all cases applicable, in addition to the usual random initialisation used for dense layers in domain adaptation. For baselines, we report results from the pre-trained models, as well as a new model with the same architecture trained from scratch (i.e. re-trained) solely on our dataset.

The input & target representations, and the optimizer used are as originally reported [15], with tabla audios upsampled to 44.1 kHz. We expect the reduced bandwidth of our data to influence the perfomance minimally since the 8-15 kHz band (which is only faintly energetic in BD and SD onsets) accounts for a minor fraction of the bins in the log-scaled spectral representation. Dropout ($p = 0.5$) is added before the first dense layer, batch size is increased to 64, and early stopping with a patience of 10 epochs is included. Learning rates are not decayed across epochs and were empirically determined to be: $1e^{-5}$ in re-training, $1e^{-6}$ for dense and and a lower $1e^{-7}$ for conv layers in differential FT (to better preserve the generalization capabilities of lower layers), and $1e^{-6}$ in all other experiments.

3.1.2 1-way Classification

The general model architecture used for this method appears in Figure 3. First, a common CNN model architecture ('Baseline' in Table 3) is obtained for all stroke classes by making modifications, targeted at achieving better convergence, to a previous architecture from ADT [16]. The input to the baseline model is a set of 3 log-scaled mel-spectrograms of dimensions 80 bands x 15 frames (150 ms) as proposed previously [26], computed from 16 kHz audios. Target activations are prepared by assigning a value of 1 to every ground truth onset frame as well as an adjacent frame on either side, and 0 to the remaining frames.

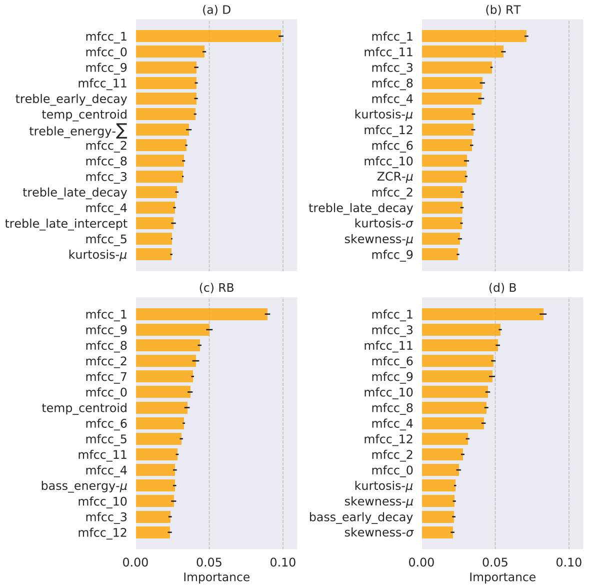

Figure 4: The 15 highest ranked features from the tabla identification task for each stroke type. Bars and whiskers show mean and standard deviation across 3 repetitions.

Models are trained to minimize BCE loss for a maximum of 150 epochs, using the Adam optimizer with *lr* equal of $1e^{-4}$, a batch size of 256, and early stopping with a patience of 10 epochs.

Considering the different amounts of data available for each stroke type, we experiment with the input representation and model architecture to arrive at the best hyperparameters for each class. A list of the chosen hyperparameter settings, along with the baseline, appears in Table 3. The input-related choices are based on experiments in ADT involving larger context [19] and alternate input representations [16]. Architecture variations to the conv and dense layer sizes are targeted at increasing the model capacity to benefit classes with more data.

3.1.3 *Random Forest Classification*

The random-forest based tabla stroke classification system presented in [12] is re-trained on our newly configured dataset to use as a baseline. This system uses the high-frequency content algorithm [27] to first segment tabla audios based on the detected onsets, and then extracts a set of 49 acoustic features for the 4-way stroke classification with a random forest. Features relating to the temporal characteristics of decay portions are found to be especially important to stroke identity across instruments.

3.2 Data Augmentation Methods

With data augmentation, we seek to simulate tabla instrument diversity from different physical structures, tuning, and playing styles. We employ pitch-shifting and time-scaling (audio-specific), attack-remixing (percussion-specific), and two methods closely tied to the acoustics and sound production characteristics of tabla - spectral filtering and NMF-based stroke-remixing (tabla-specific). All transformations are applied to the time-domain audio signal. To ensure that the modified audio sounds realistic, we enlist the help of an expert tabla player to determine the suitable range for the control parameter in each transformation. With every augmentation method, we use 4 values of the parameter (as in Table 4), obtaining 4 versions for each audio. This is combined with the original dataset, thus increasing the size to five times the original. During training, data from two CV folds (along with any augmentations) is used, while only the original data from the third fold is used for validation.

3.2.1 *Pitch-shifting (PS) and Time-scaling (TS)*

We use the *hptsm* algorithm from the python library *pytsmod* [28], which first separates an audio into harmonic and percussive components and then applies appropriate time-scaling methods to each component. Pitch-shifting is performed by time-scaling followed by re-sampling. The parameters α_{ps} and α_{ts} denote the pitch-shifting (in semitones) and time-scaling factors.

3.2.2 *Attack-remixing (AR)*

Attack-remixing refers to modifying the relative levels of attack and decay regions of an audio, and has been used to augment drums data [16]. Our implementation involves first applying harmonic-percussive separation (HPS) [29] to the audio, which leaves all the attacks in the percussive component and resonant decay portions in the harmonic component. The percussive component is scaled by a linear factor, denoted by α_{ar}, and remixed with the unmodified harmonic component.

3.2.3 *Spectral Filtering (SF)*

Augmenting data by perturbing ‘nuisance attributes’ that are unimportant in the specific discrimination task, can be effective [30]. We use the feature ranking of the random forest classifier to identify acoustic features that capture instrument characteristics and are less important to stroke identity. The classifier, as in Section 3.1.3, is re-purposed to solve a 10-way tabla identification task on our training set (spanning 10 instruments), using the same features. It achieves an average f-score of over 0.9 in a random 3-fold CV performed separately with each stroke category. From the resulting feature importances presented in Figure 4, we see that various MFCCs (computed frame-wise and averaged across stroke segments), representing spectral shape, are most important to instrument differentiation, with MFCC-1, representing the tilt (balance) between high and low frequencies, consistently at the top. This motivates the use of particular filtering transformations to modify spectral shape for the data augmentation.

Due to the contrasting broadband and band-limited nature of tabla attack and decay spectra, it is more effective to use filters targeting specific bands, instead of the commonly used random filtering [31]. After first applying HPS, we filter the bass ($0 - 200$ Hz) and treble ($200 - 2k$ Hz) regions in the harmonic component, and modify the spectral tilt in the percussive component by changing the

Method	Values	Method	Values
α_{ps}	-1, -0.5, 0.5, 1	$\alpha_{\text{sf-tilt}}$	0.2, 0.5, 2, 3
α_{ts}	0.8, 0.9. 1.2, 1.3	$\alpha_{\text{sr-bass}}$	0.6, 0.8, 1.5, 2
α_{ar}	0.3, 0.5, 2, 3	$\alpha_{\text{sr-treble}}$	0.5, 0.8, 1.5, 2
$\alpha_{\text{sf-bass}}$	0.2, 0.5, 2, 4	$\alpha_{\text{sr-damp}}$	0.2, 0.5, 2, 3
$\alpha_{\text{sf-treble}}$	0.2, 0.5, 2, 4		

Table 4: Parameter values for the augmentation methods.

energy balance across the two halves of the spectrum. The filter is a *Hann* window positioned over the corresponding band and scaled by a linear gain factor, and is multiplied with each short-time spectral slice. Each filtering operation (bass, treble, & tilt) is considered a separate transformation with the corresponding gain factor denoted by $\alpha_{\text{sf-bass}}$, $\alpha_{\text{sf-treble}}$, and $\alpha_{\text{sf-tilt}}$. We also consider a combination of all three transformations applied simultaneously by randomly selecting a value for each parameter, while still obtaining 4 augmented versions per audio.

3.2.4 Stroke Remixing (SR)

Given that the compound bols of the tabla are produced by the independent, though simultaneous, striking of the two drums, we simulate the expected variations of the relative strengths of the drums in this mix using non-negative matrix factorization (NMF). We use the *NMFToolbox* [32] to perform the decomposition. The activations are randomly initialised while the templates, a total of 6, computed separately from attack and decay regions for each of the 3 distinct stroke types (resonant bass, resonant treble, and damped), are kept fixed. Each template is the average spectrum of the corresponding portion of the signal from across 10 isolated instances. A separate set of templates is computed for each tabla instrument in our training set and used to decompose recordings from the corresponding tabla.

To perform augmentation, we first obtain the audio for each component from the decomposition, combine the attack and decay portions for each stroke type, and then resynthesize audio by mixing the three stroke components at different linearly scaled levels. We experiment with restricting scaling to only one of the three components at any time (factors denoted by $\alpha_{\text{sr-bass}}$, $\alpha_{\text{sr-treble}}$, and $\alpha_{\text{sr-damp}}$), as well as a combination with all components simultaneously scaled by different randomly chosen factors (similar to the combination method in filtering).

4. EXPERIMENTAL RESULTS

We evaluate performance using the f-score metric with a tolerance of 50 ms for the detected onset locations [33]. Scores are obtained separately for each stroke class on individual tracks and averaged across the dataset. The reported CV scores are the mean across 3 folds. For the network predictions, local peaks in the output layer activations are detected and thresholded. The threshold is selected based on maximizing validation set f-score and then used on the test set. In the transfer learning experiments, all 4 pre-trained models are tuned separately and used together (ensemble) by averaging their predicted activations during cross-validation. On the test set, an ensemble of 12 models (4 from each CV split) is utilised. With 1-way classification, single models are evaluated during CV and an ensemble of the 3 models is used on the test set.

Model	Stroke category			
	D	RT	RB	B
Baseline	84.6	83.2	**46.5**	**83.8**
↑context	84.3	81.4	41.9	73.0
Mid-channel	84.7	81.7	42.1	75.6
↑conv filters	84.7	**84.5**	44.7	77.6
↑dense units	**86.7**	82.9	40.1	73.6
↑conv filters+↑dense units	83.5	83.4	43.3	82.0
2x conv layers	84.3	82.4	42.4	75.9

Table 5: CV f-scores of 1-way model tuning experiments (bold values are highest in the column).

Method	Stroke category				Mean
	D	RT	RB	B	
No aug.	86.7	84.5	46.5	83.8	75.4
Pitch-shift	87.2	**85.5**	51.2	83.9	76.9
Time-scale	88.2	85.0	50.2	82.2	76.4
Attack-remix	84.3	84.2	48.1	81.3	74.5
SF-bass	84.5	80.9	40.4	79.9	71.4
SF-treble	85.8	81.7	48.7	76.0	73.0
SF-tilt	86.3	82.7	43.8	82.0	73.7
SF-all	87.6	84.6	50.7	85.6	77.1
SR-bass	86.0	84.8	43.3	83.6	74.4
SR-treble	86.1	84.8	39.4	79.0	72.3
SR-damp.	86.2	85.3	50.1	86.5	77.0
SR-all	86.8	85.3	48.1	84.4	76.2
Combined	**88.5**	84.2	**53.6**	**87.9**	**78.5**

Table 6: Comparing the CV f-scores of 1-way models trained using different augmentation methods (bold values are overall highest in column, underlined are top 4 among individual methods). Combined refers to PS+TS+SF-all+SR-all.

1-way model tuning: The cross-validation results of the tuning experiments with the 1-way models (discussed in Sec. 3.1.2) appear in Table 5. The input-related modifications do not lead to improved scores in any class, indicating that a 3-channel representation with moderate context duration (150 ms) is optimum for our task. With respect to model architecture, the baseline appears to be best for classes with least data (RB and B). The use of more dense units benefits only the more abundant damped class. With increased conv layer filters, the f-score for resonant treble goes up, possibly by better learning its rich and diverse harmonic content stemming from tabla tuning variations. The other modifications do not offer any further improvements.

Data augmentation: Table 6 shows the results of training the 1-way models (with hyperparameters for each class as identified in the tuning experiments), using the various augmentation methods. The underlined values are the 4

	Method	Stroke category				Mean
		D	RT	RB	B	
	Random forest	86.2 / 74.2	77.7 / 75.0	39.7 / 35.3	73.6 / 41.5	69.3 / 56.5
3-way	Pre-trained (PT)	36.8 / 27.3	15.1 / 9.0	9.8 / 19.8	7.3 / 2.1	17.3 / 14.6
	Re-trained	81.0 / 65.5	53.7 / 72.6	15.7 / 22.9	63.0 / 60.0	53.4 / 55.3
	FT dense random init.	74.4 / 65.2	55.9 / 75.2	33.6 / 45.6	63.4 / 52.0	56.8 / 59.5
	FT dense PT init.	71.7 / 62.4	54.8 / 74.9	29.4 / 34.8	60.9 / 39.1	54.2 / 52.8
	Uniform FT all	76.3 / 65.1	59.7 / 79.1	29.5 / 43.3	65.3 / 58.6	57.7 / 61.5
	Differential FT all	72.5 / 63.4	58.7 / 77.9	30.0 / 41.2	63.5 / 49.2	56.2 / 57.9
	Disjoint FT all: dense rand. init.	77.2 / 67.2	57.4 / 73.1	33.0 / **49.1**	65.9 / 60.9	58.3 / 62.6
	Disjoint FT all: dense PT init.	74.8 / 65.4	66.4 / 77.4	34.7 / 47.5	66.5 / 56.8	60.6 / 61.8
1-way	No aug.	86.7 / 79.5	84.5 / 84.1	46.5 / 38.0	83.8 / 69.0	75.4 / 67.6
	Best aug.	**88.5 / 83.3**	**85.5 / 84.3**	**53.6** / 34.1	**87.9 / 80.1**	**78.9 / 70.4**

Table 7: F-scores (CV/test) from 3-way models compared with the best 1-way models and a random-forest baseline [12] (values in bold are highest in the column). 'Best aug.' represents pitch-shifting for RT and combined aug. for the rest.

highest scores within each class that are better than no augmentation. We note that these are most often from using one of PS, TS, SF-all, SR-damp, or SR-all. We therefore experiment with combining PS, TS, SF-all, and SR-all (which includes SR-damp), by randomly choosing only 2 out of 4 versions from each method for every audio (to limit training time), taking the dataset size to 9x original. Values in bold indicate the highest scores obtained in each stroke category across all methods (individual and combination). Overall, we see that except for the resonant treble class, the combination results in the best f-scores, demonstrating the benefit of the proposed augmentation methods.

Some notable observations about the individual methods follow. The improvements from pitch-shifting and time-scaling underscore the importance of addressing tuning diversity and capturing a wide tempo range when working with datasets of realistic playing. With tabla-specific filtering and remixing, the combinations SF-all & SR-all, which pack more diversity, outperform the corresponding individual methods in most cases, and consistently give better f-scores than no augmentation.

3-way vs 1-way: Table 7 compares the CV and test set scores of the 3-way models against the best 1-way models and the random forest baseline. In the transfer learning experiments, we note that tuning conv layers helps, possibly compensating for low level acoustic differences between tabla strokes and drums. Of the three approaches to this, disjoint FT gives higher CV and test scores when compared to the other two. With regards to random versus pre-trained initialisation for dense layers in the disjoint FT setup, better test set scores are obtained with random initialisation, indicating better generalization, while PT initialisation gives higher CV scores. Finally, these domain-adapted models outperform the pre-trained only and the re-trained (from scratch) 3-way models.

Eventually, we find 1-way models mostly surpassing the best 3-way model, with data augmentation further enhancing performance. Test scores are lower than that of CV train by a few percentage points, attesting to the persistent mismatch from playing style. Only for the test resonant bass, the f-score is highest using disjoint FT, which shows that transfer learning has helped with generalization for the class with least data. A closer look at disjoint FT versus the 1-way models further reveals that the most difference in f-score is in resonant both, indicating that treating the combination stroke as a separate class works better than viewing it as the superposition of its component stroke classes. Finally, it is interesting to note that the 3-way models trained from scratch perform much poorer than the set of similarly trained 1-way models, demonstrating the benefit of using separate models specialised for each class in this task.

5. CONCLUSIONS

We presented a four-way tabla stroke classification task for categories defined by the salient acoustic characteristics of the basic tabla strokes. Leveraging the similarity of our target categories with popular Western drumkit classes, we investigated methods from the automatic drums transcription task. We explored the adaptation of available pre-trained drums models via transfer learning on a new tabla dataset. Systematic experiments with different transfer learning strategies reveal significant improvements when both dense (classifier) layers and conv (feature extractor) layers of a multiclass CNN model are fine-tuned from the pre-trained weights in a disjoint fashion. Next, the use of separate 1-way CNN models with hyperparameters suitably tuned for each of the 4 stroke categories was found to surpass the more complex 3-class CNN model for all class accuracies except the most data-constrained resonant bass category, which benefited from pre-training on drums. Further, several data augmentation methods, untested so far in the context of tabla, were investigated. A method based on increasing training data diversity, by varying spectral characteristics that capture instrument-dependence across strokes, contributed consistently to improved classification accuracy. Future work will target recurrent architectures and the combination of transfer learning and data augmentation for further performance gains.

Supplementary: github.com/DAP-Lab/4way-tabla-transcription

6. REFERENCES

[1] P. Chordia, "Automatic transcription of tabla music," Ph.D. dissertation, Stanford University, 2006.

[2] O. Gillet and G. Richard, "Automatic labelling of tabla signals," in *Proc. of the 4th Int. Society for Music Information Retrieval Conf.*, Baltimore, U.S.A., 2003.

[3] P. Chordia, "Segmentation and recognition of tabla strokes," in *Proc. of the 6th Int. Society for Music Information Retrieval Conf.*, London, U.K., 2005.

[4] S. Gupta, A. Srinivasamurthy, M. Kumar, H. A. Murthy, and X. Serra, "Discovery of syllabic percussion patterns in tabla solo recordings," in *Proc. of the 16th Int. Society for Music Information Retrieval Conf.*, Málaga, Spain, 2015.

[5] S. Deolekar and S. Abraham, "Tree-based classification of tabla strokes," *Current Science (00113891)*, vol. 115, no. 9, pp. 1724–1731, 2018.

[6] R. Sarkar, A. Singh, A. Mondal, and S. K. Saha, "Automatic extraction and identification of bol from tabla signal," in *Advanced Computing and Systems for Security: Volume Five*, R. Chaki, A. Cortesi, K. Saeed, and N. Chaki, Eds. Springer Singapore, 2018, pp. 139–151.

[7] S. Shete and S. Deshmukh, "Analysis and comparison of timbral audio descriptors with traditional audio descriptors used in automatic tabla bol identification of north Indian classical music," in *Proc. of the 20th Int. Conf. on Computational Science and Applications*, S. Bhalla, P. Kwan, M. Bedekar, R. Phalnikar, and S. Sirsikar, Eds. Springer Singapore, 2020, pp. 295–307.

[8] ——, "North Indian classical music tabla tala (rhythm) prediction system using machine learning," in *Advances in Speech and Music Technology*, A. Biswas, E. Wennekes, T.-P. Hong, and A. Wieczorkowska, Eds. Singapore: Springer Singapore, 2021.

[9] M. Clayton, *Time in Indian Music: Rhythm, Metre, and Form in North Indian Rag Performance*. Oxford University Press, U.K., 2001.

[10] D. Courtney, *Fundamentals of Tabla*. Sur Sangeet Services, 2013.

[11] M. Clayton, "Theory and practice of long-form non-isochronous metres," *Music Theory Online*, vol. 26, no. 1, 2020.

[12] M. A. Rohit and P. Rao, "Automatic stroke classification of tabla accompaniment in hindustani vocal concert audio," *To appear in Journal of Acoustical Society of India*, April 2021. [Online]. Available: https://arxiv.org/abs/2104.09064

[13] P. Herrera, A. Yeterian, and F. Gouyon, "Automatic classification of drum sounds: A comparison of feature selection methods and classification techniques," in *Proc. of the 2nd Int. Conf. on Music and Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 69–80.

[14] C. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, "A review of automatic drum transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, 2018.

[15] R. Vogl and P. Knees, "Mirex submission for drum transcription 2018," in *Proc. of the 19th Int. Society for Music Information Retrieval Conf.*, Paris, France, 2018.

[16] C. Jacques and A. Röbel, "Automatic drum transcription with convolutional neural networks," in *Proc. of the 21th Int. Conf. on Digital Audio Effects*, Aveiro, Portugal, 2018.

[17] ——, "Data augmentation for drum transcription with convolutional neural networks," in *Proc. of the 27th IEEE European Signal Processing Conf.*, A Coruña, Spain, 2019.

[18] R. Gowriprasad and K. S. R. Murty, "Onset detection of tabla strokes using LP analysis," in *Proc. of the 13th IEEE Int. Conf. on Signal Processing and Communications (SPCOM)*, Bangalore, India, 2020.

[19] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, "Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks," in *Proc. of the 18th Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017.

[20] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new python audio and music signal processing library," in *Proc. of the 24th ACM Int. Conf. on Multimedia*, 2016.

[21] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. of the 18th Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017.

[22] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," in *Proc. of the 44th IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Brighton, U.K., 2019.

[23] A. Diment and T. Virtanen, "Transfer learning of weakly labelled audio," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, U.S.A., 2017.

[24] S. Zhang, C.-T. Do, R. Doddipatla, and S. Renals, "Learning noise invariant features through transfer learning for robust end-to-end speech recognition," in *Proc. of the 45th IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020.

[25] P. G. Shivakumar and P. Georgiou, “Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations,” *Computer Speech & Language*, vol. 63, 2020.

[26] J. Schlüter and S. Böck, “Improved musical onset detection with convolutional neural networks,” in *Proc. of the 39th IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Florence, Italy, 2014.

[27] P. Brossier, J. P. Bello, and M. D. Plumbley, “Fast labelling of notes in music signals,” in *Proc. of the 5th Int. Society for Music Information Retrieval Conf.*, Barcelona, Spain, 2004.

[28] S. Yong, S. Choi, and J. Nam, “Pytsmod: A python implementation of time-scale modification algorithms,” in *Extended Abstracts for the Late-Breaking Demo Session of the 21st Int. Society for Music Information Retrieval Conf.*, Montréal, Canada, 2020.

[29] D. FitzGerald, “Harmonic/percussive separation using median filtering,” in *Proc. of the 13th Int. Conf. on Digital Audio Effects*, Graz, Austria, 2010.

[30] W.-N. Hsu, Y. Zhang, and J. Glass, “Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation,” in *IEEE Automatic Speech Recognition and Understanding Workshop*, Okinawa, Japan, 2017.

[31] J. Schlüter and T. Grill, “Exploring data augmentation for improved singing voice detection with neural networks,” in *Proc. of the 16th Int. Society for Music Information Retrieval Conf.*, Málaga, Spain, 2015.

[32] P. López-Serrano, C. Dittmar, Y. Özer, and M. Müller, “NMF toolbox: Music processing applications of nonnegative matrix factorization,” in *Proc. of the 22nd Int. Conf. on Digital Audio Effects*, Birmingham, U.K., 2019.

[33] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “mir_eval: A transparent implementation of common MIR metrics,” in *Proc. of the 15th Int. Society for Music Information Retrieval Conf.*, Taipei, Taiwan, 2014.