# FOLDED CQT RCNN FOR REAL-TIME RECOGNITION OF INSTRUMENT PLAYING TECHNIQUES

Jean-Francois Ducher, Philippe Esling

# FOLDED CQT RCNN FOR REAL-TIME RECOGNITION OF INSTRUMENT PLAYING TECHNIQUES

**Jean-François Ducher**
CICM – MUSIDANSE –
Université Paris 8/
IRCAM (UMR9912 STMS)
ducher@ircam.fr

**Philippe Esling**
IRCAM (UMR9912 STMS)
esling@ircam.fr

## ABSTRACT

In the past years, deep learning has produced state-of-the-art performance in timbre and instrument classification. However, only a few models currently deal with the recognition of advanced *Instrument Playing Techniques* (IPT). None of them have a real-time approach of this problem. Furthermore, most studies rely on a single sound bank for training and testing. Their methodology provides no assurance as to the generalization of their results to other sounds. In this article, we extend state-of-the-art convolutional neural networks to the classification of IPTs. We build the first IPT corpus from independent sound banks, annotate it with the JAMS standard and make it freely available. Our models yield consistently high accuracies on a homogeneous subset of this corpus. However, only a proper taxonomy of IPTs and specifically defined input transforms offer proper resilience when addressing the "*minus-1db*" methodology, which assesses the ability of the models to generalize. In particular, we introduce a novel *Folded Constant Q-Transform* adjusted to the requirements of IPT classification. Finally we discuss the use of our classifier in real-time.

## 1. INTRODUCTION

Throughout modern history, western composers have diversified and refined *Instrument Playing Techniques* (IPTs) in order to foster innovation in the timbre space [14]. In folklore and oral traditions, IPTs sometimes stand out as a distinctive feature of the musical style [16, 22]. Therefore, their identification could contribute to the more general task of style recognition in the process of browsing in music databases. Moreover, interactive computer music systems (for instance in the field of improvisation or score-following) could hugely benefit from the development of real-time IPT classifiers [24].

In the last two decades, the MIR community has produced a lot of research in the field of timbre recognition, but there has been little effort in IPT classification, often considered as its *last frontier* [17].

One major cause of this gap in research is the lack of IPT sound banks. Lostanlen [17] has recently addressed the question of IPT recognition but limited his experiment to samples from isolated notes in a unique sound bank.

Here, our aim is to build a real-time classifier of IPT from solo recordings. Our system should be reactive to possibly rapid changes in the technique. Therefore, the preprocessing of the audio has to maintain temporal coherence and induce as little latency as possible. For instance, any segmentation of the audio (such as proposed by [21]) in order to subsume our task into a problem of classification of isolated notes would be irrelevant. Our study focuses on the cello but the methodological issues we raise are similar for other instruments and the process we use to build and train the classifier could be generalized as long as the IPTs of these instruments are included in a sufficient number of sound banks.

We show that trying to categorize cello IPTs in a unidimensional way produces weak results. The classifier performs well on homogeneous sets of data but generalizes poorly. Therefore we introduce a taxonomy of the playing techniques of the cello along 4 axes (named *exciter/vibrator*, *left-hand*, *waveform*, and *interaction position*). We aim to build a single network which classifies audio sequences in a multi-task [4] manner according to these axes. Then, we implement a rule-based system on top of this network, in order to simplify the model and yield a classification along the 18 main IPT categories.

In order to train our classifier, we produce a large corpus of labeled synthetic data with 5 IPT sound banks and their proprietary samplers. This corpus is annotated using the JAMS standard [11]. We make it available to the MIR community.

We adapt state-of-the-art models successfully used for instrument classification [11,16] to the multi-task and low latency requirements of IPT recognition. Front-end classifiers along the 4 IPT dimensions are built as fully-connected (FC) or recurrent layers on top of deep convolutional neural networks (CNNs). All tested system configurations achieve high accuracies on homogeneous subsets of our annotated corpus. This alone provides no indication of their ability to generalize to other databases or actual solo recordings. Therefore, we adapt the *minus-1db* methodology presented by Livshin [15] to the needs of our system. When subject to this methodology, we

**Attribution**: Jean-François Ducher, Philippe Esling. "Folded CQT RCNN for Real-time Recognition of Instrument Playing Techniques". 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

show that RNN front-ends generalize better than FC. Our adapted *Folded Constant-Q Transform (FCQT)* also yields more stable performance than Log-Mel-Spectrograms. Finally, we assess the reactivity of our models for each of the 4 IPT dimensions.

## 2. RELATED WORK

### 2.1 Deep Learning and MIR

Following their success in the field of computer vision [10], deep learning techniques have been quickly adopted by the MIR community. Instead of using sets of hand-designed audio descriptors [1,7,15], these techniques rely on basic representations of the audio signal and let the algorithm learn suitable features for a given task. Convolutional (CNN), recurrent (RNN) neural networks and their combinations have been among the most popular architectures used in MIR. Convolutional layers seek local correlations within their input by training sets of convolutional kernels. CNNs are built by stacking such layers with *pooling* layers[1] at increasingly bigger scales. Therefore, they can detect large and complex patterns while being computationally efficient. RNNs have been developed in order to forecast or classify temporal sequences. As their hidden units have connections from one time step to the next, they can carry information through various temporal states. Long Short-Term Memory (LSTM) units, where gates enable to control this flow of information, have been proficiently used in MIR [6].

### 2.2 Instrumental Timbre Classification

Early research in instrument classification relied on samples of isolated notes played with ordinary techniques. In most studies, experiments were performed with a single sound bank. This practice overlooked variability related to the instrument model, player or recording environment. A detailed review of generalization issues by Livshin [15] shows that the performance of classifiers trained and tested with a single sound bank gives no hint on their accuracy when confronted to new sounds. He suggests to use several independent sound banks, pick one for testing while training the classifier on the rest joined together. Then, the experiment has to be repeated with all the possible test banks. This methodology named *minus-1db* provides more reliable indications on the ability of the classifier to generalize. In the case of instrument classification from solo recordings, it translates into a *leave-1-CD-out* policy.

To our knowledge, very few studies follow the methodological principles of Livshin, and, as such, can be regarded as being state-of-the-art in this matter.

Patil and al. [21] proposed a classifier built upon a support vector machine applied to spectro-temporal receptive fields. Trained on isolated notes of the RWC database to classify 6 instruments, this model reached 98.7% accuracy. Its resilience was assessed on a proprietary database of soli, which were first segmented using a harmonicity-based method. With the *leave-1-CD-out* methodology, accuracy still reached 88.1%.

Lostanlen and Cella [18] used two separate solo instrument databases to train and test a deep CNN to classify 8 instruments. Their network relied on the CQT of the audio signal. Through proper optimization of their convolution strategy, their system reaches average accuracies of 74%, against 61.4% for a decision tree forest applied to a large set of audio descriptors.

Regarding predominant instrument classification in polyphonic textures, Han and al. [11] achieved state-of-the-art F1-scores with a deep CNN applied to log-mel-spectrograms. This study was performed with two independent subsets of the IRMAS database for training and testing.

### 2.3 IPT Classification

IPT classification studies have been carried out on the clarinet [19], the snare drum [27], and the electric guitar [5]. The first two studies pose methodological issues since they perform training and evaluation with a single database. Chen and al. [5] focus on the detection in electric guitar solos of five techniques which all have an impact on the melodic contour. This feature is key to the design of their classifier. Therefore, their research can hardly be generalized to other IPTs.

Lostanlen and al. [17] tackle the issue of IPT recognition in a transversal manner. They work with samples from isolated notes belonging to 143 IPTs from 16 instruments. Their query-by-example system relies on a variant of the *k-nearest neighbors* algorithm where the metric used is subject to a training process. Applied to Mel-Frequency Cepstral Coefficients enriched by second-order scattering coefficients, it reaches *rank-5* accuracy of 61%. Yet, again, they train and test their system on a single sound bank.

## 3. IPT TAXONOMY

As we will show in Section 5., trying to classify IPTs without taking into account their multi-dimensional structure results in a poor ability to generalize. This motivates our newly introduced taxonomy.

### 3.1 Theoretical Background

A proper IPT taxonomy requires identifying what exciter, vibrator and resonator are selected (Schaeffer [26]), and what modification and excitation movements are undertaken (Cadoz [3]). Taking the example of the cello and following Feron [8] :

- among the possible exciters are the bow hair, bow wood, as well as various parts of the hand (finger, nail, knuckle). The natural vibrators are the four strings, but the body can be involved[2];
- modification movements are mainly the ornaments and other IPTs realized by the left hand (e.g. *vibrato*, *glissandi*, *harmonics*);

[1] Pooling reduces the size of the output of the convolution (*called a feature map*) by downsampling ; generally, the maximum value of a local neighborhood is taken (*max-pooling*)

[2] The *resonator* is assumed to remain unchanged (body).

– excitation movements should be characterized by their position (e.g. *sul tasto* or *ponticello),* length (e.g. *staccato*), eventual periodicity (e.g. *jettato, tremolo*), and the amount of speed and pressure involved (e.g. *flautando* vs. *pressured*).

### 3.2 Availability of Data

Proper definitions of all these IPTs should then be provided in order to annotate a recorded corpus of audio in a consistent manner.

However, the cost of such a study would be prohibitive. Therefore, we decided to rely upon available sound banks and their IPT definitions.

We identified 5 IPT sound banks which had different players and recording setups: EastWest Quantum Leap (EWQL), Vienna String Library (VSL), IRCAM Solo Instruments (ISI), Virtual Orchestra (VO) and ConTimbre (CONT). These banks suffer from two drawbacks. First, as expected, the absence of standardized definitions causes gaps in the realization of given IPTs between them, even for such a basic feature as the vibrato of an *ordinario* class. Second, each of the sound banks includes only a fraction of all the technical possibilities mentioned above. Several IPT combinations, albeit perfectly playable, are not available (e.g. *harmonic trills*).

### 3.3 Proposed Taxonomy

We match the list of IPTs in our sound banks with the theoretical approach in section 3.1. Bearing in mind our real-time constraint, we want to prevent an inflation of the number of classifiers and parameters in our model.

Therefore, we retain only 4 dimensions in our taxonomy (see Table 1). The first axis refers to the *exciter/vibrator* couple, which has a strong impact on the harmonicity and noisiness of the resulting sound. The second axis refers to how the *left hand* shapes the pitch contour. The third axis, called *waveform*, classifies IPTs depending on the nature and length of the bow/string interaction. The last axis refers to the *position* of the *interaction* with the string, which induces different spectral envelopes in the sound.

| Axis 1<br>*Exciter/Vibrator* | Axis 2<br>*Left-hand* | Axis 3<br>*Waveform* | Axis 4<br>*Int. Position* |
|---|---|---|---|
| bow hair/string (*ordinario*) | NONE | NONE | NONE |
| bow wood/string (*con legno*) | *vibrato* | sustained | ordinario |
| finger/string (*pizzicato*) | non *vibrato* | *staccato* | *sul tasto* |
| finger/string+body (*pizz. Bartok*) | *glissando* | *spiccato, battuto* | *sul ponticello* |
| pressured bow/string | trill | *marcato, sfz* | harmonics |
| hand or knuckle hit on body | | *tremolo* | |

**Table 1.** IPT taxonomy (axes, classes) proposed in this study. This taxonomy is partly hierarchical in the sense that classification along Axes 2-4 is optional and dependent upon classification on Axis 1. When no classification is desirable, the *NONE* class is used in the training process[3].

Some IPTs which would require a separate axis (should we aim at an exhaustive taxonomy for musicological purposes) were forced onto an existing dimension. Pressured bowing was included in axis 1 as it results in strong inharmonicity and noisiness. Harmonics, both natural and artificial, were included in axis 4, as they are most commonly played *sul ponticello*, but imply specific spectral envelopes. Finally, string classification[4] was excluded from the taxonomy, as well as the use of mutes.

On each dimension, each class of the taxonomy had to be represented at least in two sound banks for the *minus 1-Db* methodology to be implemented.

## 4. EXPERIMENTS

### 4.1 Building the Databases

#### 4.1.1 *Sequence Generation Principles*

Bearing in mind our goal to generalize to actual solo recordings, we have generated series of audio sequences which simulate such recordings. This simulation tool was developed as a series of Max/Msp patches[5] controlling the proprietary samplers[6] of our 5 sound banks. The generated sequences include randomly generated notes and chords of all available IPTs. The chords are bounded by the playability of the instrument (as presented in [28]) and the specific ranges of the sound banks.

In the remainder of this article, we will use the term *database X* to designate the sequences which were generated with this process applied to the *sound bank X.*

#### 4.1.2 *Data Augmentation*

We augment the data to increase the robustness of the classifier to variations in the recording environment and tuning of the instrument. Therefore, we generate sequences with a randomly detuned reference A4 in a 20Hz range around 440Hz (through transposition of the original samples). We also use various levels and types of reverb in the samplers. Finally, we average the stereo channels provided by the sampler to get the final signal.

After augmentation, the sequences represent 13.5 hours of music and over 4 Gb of AIF files sampled at 44100Hz and encoded at 16 bits PCM.

#### 4.1.3 *JAMS Standard Annotation*

Our simulation tool exports both the annotation files with the JAMS format [13] and the corresponding audio. IPT classification along the 4 axes is provided with the *tag_open* namespace. Onset times and note/chord pitches were added when available, using *onset* (resp. *pitch_contour*) namespaces. Both audio and annotation files are made available to the MIR community[7].

[3] For instance, a pizzicato will never be classified along the 3rd (waveform) axis. But it still could be classified along the 2nd axis (e.g. glissando) or the 4th (e.g. harmonics).

[4] All notes above G2 can be played on several strings. String change is regarded as a component of intra-class variability.

[5] Available upon request

[6] e.g. UVI Workstation for ISI, Vienna Instruments for VSL.

[7] https://drive.google.com/open?id=1HYqHxxd2ZDkU2TL_1EXa6WNv9lY37hU9

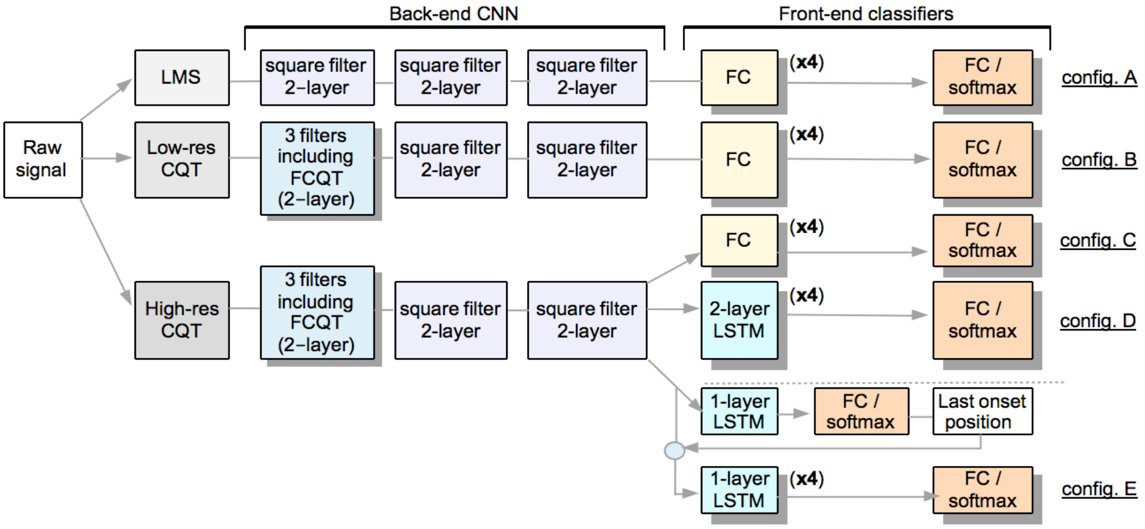

**Figure 1.** System architecture (incremental configurations A to E). A is directly inspired by [11]. B introduces CQT along with adapted filtering in the convolutional process. The resolution of CQT is increased in C. Recurrent layers process the output of the CNN in D. Finally, in configuration E, we experiment with the joint classification of onsets.

## 4.2 System Architecture and Configuration

### *4.2.1 Preprocessing*

As shown in Figure 1, we have tested several possibilities[8] for the spectral transform.

*Log-mel-spectrograms (LMS)*: following [11], we first downsample the signal to 22,050Hz, then compute LMS on 128 bins, with FFT windows of 2048 samples and hopsize of 512 samples (~23ms).

*Adapted low-res CQT*: following [18], we compute a 12 bin-per-octave CQT from C3(~130Hz)[9] to B10(~15.8kHz). Hop size is 1024 samples (~23ms). Only the logarithm of the amplitude of the CQT is retained. In order to preserve the temporal coherence of the preprocessed signal, we halve the Q-factor of the bins in the C3-B3 octave. In our adapted CQT, the size of the analysis windows are limited to 2850 samples (~64ms).

*Adapted high-res CQT*: to better account for IPTs such as *glissandi* and *vibrato*, we also experiment with a doubled bin-per-octave resolution above C5. The total number of bins of the CQT increases from 84 to 144. Since analysis windows are bounded, we cannot extend this resolution to the lower two octaves without deteriorating further the corresponding Q-factors.

In all configurations, we filter out low-energy frames (with average LMS or CQT < -79dB), normalize the data and cut it into fixed-length sequences of 60 frames (~1.4 s). Short sequences are less likely to include the attack of long notes, which is critical information for the network. A sequence length of 60 frames is an empirical compromise between this loss of information and the computing cost of longer sequences.

### *4.2.2 Folded Constant Q-Transform (F-CQT)*

We introduce *F-CQT* as a generalization of the *pitch spiral* method [18] in order to capture efficiently the spectral envelope of a signal. It is obtained by first changing the pitch order of the CQT chromatic bins to match the cycle of fifths. The reshuffled CQT is then folded in 2 dimensions so as to put successive octaves on adjacent lines, in the same manner as the pitch spiral.

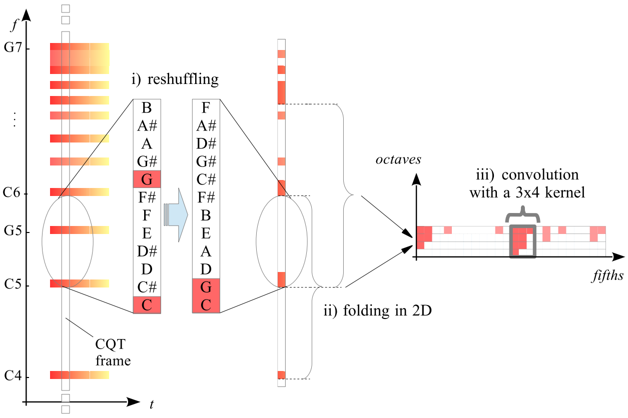

**Figure 2.** *F-CQT* example for a C4 note: (i) each CQT frame is reshuffled on an octave-per-octave basis, folded (ii) using 2-octave wide half-overlapping bands, in order to avoid side effects. A kernel (iii) of size 12 captures 8 out of the first 12 harmonics including the fundamental.

As shown in Figure 2, bins related to the harmonics $f_{2i+3j} = f_1 2^i 3^j$ of a given fundamental $f_1$ remain in its close neighborhood. Therefore, a small convolutional kernel of 3 *fifths* x 4 *octaves*[10] will capture 8 out of the first 12 harmonics. This is achieved without resorting to a computationally expensive harmonic-CQT [2]. Convolution with such a kernel can be seen as a 1D frequency-wise convolution of the usual CQT with a disjoint filter: the *F-CQT filter*. To capture the same

[8] We use the Librosa [20] implementation for LMS and CQT.

[9] It has been assumed that the lower octave of the cello would be analyzed with enough detail without the fundamental frequency being reported.

[10] Tradeoff between the number of harmonics captured and the size of the kernel. A higher number of octaves results in blurring the picture with several harmonics on the same bin.

harmonics with a regular 1D-kernel would require a much bigger kernel size (43 parameters instead of 12).

### 4.2.3 *CNN Back-end*

We assume that the nature of our task is somewhat similar to the recognition of instruments in a challenging environment such as polyphonic textures. Therefore, we follow the main characteristics of the CNN architecture presented in [11], while adapting its capacity to our data.

The proposed CNN is made of 3 modules which operate at increasing scales. In each module we stack two identical convolutional layers, with batch normalization and *Rectified Linear Unit* (ReLU) activation. A max-pooling layer and dropout at 0.25 probability are implemented at the output of the module. Following the architecture design of [11], we use as baseline square (3x3) filters[11] for all layers. However it has been suggested in recent research [23] that domain-specific filtering could improve CNN performance, especially in the deeper layers. Therefore, we evaluate the use of three separate filters, namely our *F-CQT*, the baseline (3x3) and a (6x2) filter; the latter is designed to capture longer patterns such as *vibrato* or *trills*. The concatenation of 8 feature maps for each filter is used as input of the second module. Max-pooling layers are in charge of downsampling the features while the number of feature maps increases[12]. The output of the CNN is a 10 step long sequence of 64 maps with a single feature (one step equals six frames ~125ms).

### 4.2.4 *IPT Classifiers (Front-end)*

In configurations A to C, IPT classifiers are built with a fully-connected (FC) layer of 32 neurons with *ReLU* activation, followed by another FC layer with *softmax* activation. The latter comprises as many neurons as there are classes in the corresponding axis [10].

In configuration D (resp. E), we replace the first FC layer with a double (resp. single) layer of 32 unidirectional LSTM cells with an input of 64 features per temporal step.

In configuration E, following [12], an additional classifier with the same design is jointly trained to locate the attack of the last note of the sequence. Its eleven classes coincide with the 10 steps of the sequence plus one: this additional class is used to categorize sequences of long notes where the attack occurs prior to the beginning of the sequence. The prediction of this onset classifier is concatenated with the original features and used as input to the 4 IPT classifiers.

A rule-based system computes an 18-class linear classification from the network predictions along the 4 IPT axes. The same rules are applied to the ground truth. An 18-class accuracy is provided as an additional assessment of the performance of the system.

### 4.3. Training Configuration

The system is trained by minimizing the sum of the cross-entropy loss function of the classifiers. Mini-batch gradient descent is performed with ADAM optimization [10] and exponential decay of the learning rate[13].

## 5. RESULTS

### 5.1 Direct classification (18 classes)

We first attempted to build directly a classifier among 18 cello IPTs, chosen simply because they were the most represented IPTs in our data. Our network architecture (called $A_{18}$) was similar to configuration A, but with a single 18-class classifier front-end. This effort resulted in excellent accuracies when the classifier was tested on homogeneous subsets of our corpus (see Table 2). However, these results collapsed when the *minus-1db* methodology was implemented. Not only average accuracies dropped below 50% but they were very irregular from one test base to the other. This indicated a poor ability to generalize.

| Database excluded from training | 18-class accuracy | |
|---|---|---|
| | Homogeneous corpus | Heterogeneous corpus *(minus−1db)* |
| CONT | 92,9% | 49,9% |
| EWQL | 94,0% | 30,3% |
| ISI | 95,2% | 51,1% |
| VSL | 97,3% | 44,0% |
| VO | 93,4% | 32,4% |
| Average $A_{18}$ | 94,6% | 41,5% |

**Table 2.** 18-class accuracy of configuration $A_{18}$ with a single 18-class IPT classifier front-end. Results are given for 5 different training/testing subsets and averaged on 3 alternative learning schedules.

### 5.2 Introduction of our taxonomy

Trained and tested on 5 homogeneous subsets of our data, our models still yield in average 18-class accuracies over 90% in all the configurations (see Table 3). In the framework of the *minus-1db* methodology, they now exhibit much greater resilience. Their 18-class accuracies, averaged on all test bases, are above 50% in most configurations. These accuracies are also less sensitive to the choice of the test base, as shown by Figure 3 in the case of configuration D.

| Configuration | Parameter count | 18-class accuracy | |
|---|---|---|---|
| | | Homogeneous corpus | Heterogeneous corpus *(minus−1db)* |
| A (LMS) | 184K | 93,5% | 49,6% |
| B (Low-res CQT) | 180K | 90,1% | 52,2% |
| C (High-res CQT) | 181K | 91,2% | 54,3% |
| D (2 layers LSTM) | 150K | 91,3% | 57,6% |
| E (Joint onset class.) | 142K | 91,7% | 57,6% |

**Table 3.** Parameter count and 18-class accuracy of configurations A-E, averaged on 5 different homogeneous or heterogeneous training/testing subsets and 3 alternative learning schedules.

On all axes but the *interaction position*, whatever the resolution, CQT with adapted filtering performs better than 128 bins log-mel-spectrograms (see Table 4). For

[11] 2D filters are noted: N *time frames* x M *frequency bins*.

[12] Detailed architecture available here:https://drive.google.com/open?id=1GvS6VQ3iJP6e9MBajEPL0VOXva-iuUmS

[13] Detailed training parameters available at the same URL as 12.

instance, configuration C achieves better average and 18-class accuracy than A with roughly the same parameter count (Student t-test resp. p=0.046 and 0.004).

| Accuracy / Configuration | Average on all axes | Exciter/ vibrator | Left-hand | Wave-form | Int. Position |
|---|---|---|---|---|---|
| A (LMS) | 72,6% | 84,2% | 66,6% | 77,1% | 62,7% |
| B (Low-res CQT) | 73,6% | 85,6% | 71,1% | 78,1% | 59,7% |
| C (High-res CQT) | 74,7% | 86,0% | 72,6% | 78,6% | 61,7% |
| D (2 layers LSTM) | **76,0%** | **87,0%** | 73,4% | **79,9%** | **63,5%** |
| E (Joint onset class.) | 75,6% | 86,7% | **73,8%** | 79,5% | 62,2% |

**Table** 4. *Minus-1db* framework : per-axis and average accuracies in each configuration, for all 5 test databases and 3 alternative learning schedules.

We hypothesized that increasing the resolution of the CQT beyond the tempered scale would improve system accuracy for such IPTs as *vibrato* or *glissando*. Our experiment confirms that configuration C yields better average accuracies than configuration B on the *left-hand* axis (p=0.023). This results in better 18-class accuracy (p=0.045).

Configuration D with a 2-layer LSTM front-end achieves better average and 18-class accuracy than C with a fully-connected front-end (resp. p=0.039 and 0.02). Configuration E with joint onset classification but single-layer LSTMs also achieves better 18-class accuracy than C (p=0.016) with even lower parameter count. In both configurations E and D, all axes exhibit average improved performance compared to C but detailed results show discrepancies from one test base to the other.

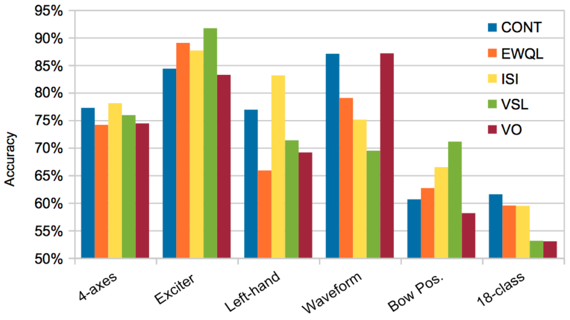

**Figure** 3. Accuracy per base and axis (configuration D).

In all tested configurations, the best accuracies are observed on the *exciter/vibrator* axis, while the worst-performing axis is *interaction position*. This statement is valid across most test databases, as seen in Figure 3 for configuration D. Bow position classification is likely to be a very difficult task even for a human expert. In the medium register, the choice of the string has a strong impact on the timbre, which makes the bow position harder to guess. In the higher register, *sul tasto* (with stronger emphasis or lower rank harmonics) and *ordinario* may be hard to distinguish. Finally, this is the axis where variability due to the instrument model is likely to be most perceivable.

### 5.3 Reactivity study

Our real-time classifier has to be as reactive as possible to sudden changes in the play of the cellist. To assess that reactivity, we select sequences in our test database where a change of IPT just occurred on the last note (or chord) of the sequence. We compute the average accuracy of the system as each time frame goes by. As Figure 4 exhibits, *exciter* is the axis where the classifier is most reactive, achieving >70% accuracy within 70ms of the attack. Unsurprisingly, it takes much longer for the network to categorize *left-hand* activity (e.g. *vibrato, trills)* or discriminate between *waveforms,* which often requires the note to be released (e.g. s*taccato*). Finally, not only the *bow-position* axis yields poorer overall accuracy, but it is also the least reactive.

When the attack of the note gets out of the 60 frame-wide analysis window (see Figure 4, right side), the system has to categorize IPTs without information about the attack. However, its performance is not harmed as one could expect. Indeed, the actual length of the note provides information about the technique used. Longer notes tend, for instance, to be produced with the bow and to be vibrated or trilled.

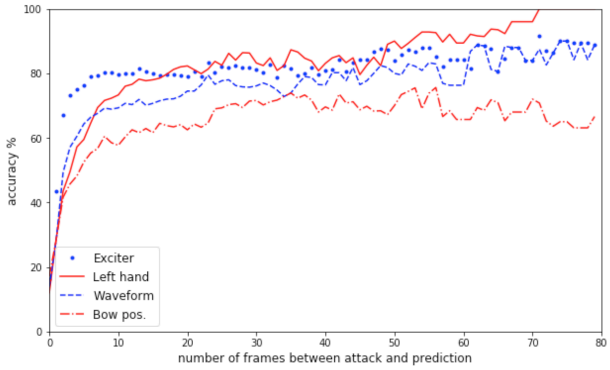

**Figure** 4. Average accuracy for sequences with an IPT change vs. time lapsed between change and prediction (10 frames~0,23s). Test base: ISI, configuration D.

## 6. CONCLUSIONS

In this article, we have extended state-of-the-art methods regarding instrument recognition to the real-time classification of IPTs from cello solo recordings. First experiments in the framework of the *minus-1db* methodology show a good resilience of models which are based on a meaningful taxonomy and process an adapted CQT through the proper combination of deep CNN and LSTM layers. Our methodology, from the realization of the databases to the architecture and training of the networks, can be extended with little effort to other string instruments. Other orchestral instruments first require an adaptation of the IPT taxonomy, which could be grounded on the same principles as ours [3,8,26].

To further assess the ability of our models to generalize, a database of contemporary cello solo recordings has been built, some of which will be annotated with the JAMS standard and used for testing. Finally, several unsupervised adaptation techniques, such as Maximum Classifier Discrepancy [25] or back-propagation through Gradient Reversal Layer [9], will also be tested in this environment.

## 7. REFERENCES

[1] D.G. Bhalke, C.B. Rama Rao, and D.S. Bormane: "Automatic Musical Instrument Classification using Fractional Fourier Transform based-MFCC Features and Counter Propagation Neural Network", *Journal Int. Inform. Syst.,* Vol. 46.3, pp. 425–446, 2016.

[2] R.M. Bittner, B. McFee, J. Salamon, P. Li, J.P. Bello: "Deep Salience Representations for $f_0$ Estimation in Polyphonic Music", *Proc. International Society for Music Information Retrieval (ISMIR)*, 2017.

[3] C. Cadoz: "Instrumental gesture and musical composition", *Proc. Int. Computer Music Conf.*, 1988.

[4] R. Caruana: "Multitask Learning". *Machine Learning*, Vol. 28, 1997.

[5] Y.-P. Chen, L. Su, and Y.-H. Yang: "Electric Guitar Playing Technique Detection in Real-World Recording Based on F0 Sequence Pattern Recognition", *Proc. ISMIR*, 2015.

[6] K. Choi, Fazekas G., K. Cho and M. Sandler: "A Tutorial on Deep Learning for Music Information Retrieval", arXiv:1709.04396v2, 2018.

[7] S. Essid: "Classification Automatique des Signaux Audio-fréquences: Reconnaissance des Instruments de Musique", PhD Thesis, Université Pierre et Marie Curie, Paris, 2005.

[8] F.-X. Féron, B. Carat: "Catégorisation des Actions Instrumentales dans Pression pour un(e) Violoncelliste de Lachenmann", *Manifeste 2014, Conférence "Composer (avec) le geste"*, https://medias.ircam.fr/x23f46d, 2014.

[9] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand and V. Lempitsky: "Domain-Adversarial Training of Neural Networks", *Journal of Machine Learning Research*, Vol. 17, p. 1-35, 2016.

[10] I. Goodfellow, Y. Bengio, and A. Courville: *Deep Learning*, MIT Press, Cambridge MA, USA, 2017.

[11] Y. Han, J. Kim, and K. Lee: "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25, no. 1, pp. 208–221, 2017.

[12] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, D. Eck: "Onsets and Frames: Dual-Objective Piano Transcription", *Proc. ISMIR*, 2018.

[13] E.J. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R.M. Bittner, J.P. Bello: "JAMS : a JSON Annotated Music Specification for Reproducible MIR Research", *Proc. ISMIR*, 2014.

[14] S. Kostka: *Materials and Techniques of Post Tonal Music,* Taylor&Francis, pp. 216-223, 2016.

[15] A. Livshin: "Automatic Musical Instrument Recognition and Related Topics. Acoustics", PhD Thesis, Université Pierre et Marie Curie, Paris VI, 2007.

[16] D.M. Lloyd: "A Classical Clarinetists Guide to Klezmer Music", PhD Thesis, Ohio State University, pp. 37-43, 2017.

[17] V. Lostanlen, J. Andén, and M. Lagrange: "Extended Playing Techniques: the Next Milestone in Musical Instrument Recognition". *5th International Workshop on Digital Libraries for Musicology*, Paris, France, 2018.

[18] V. Lostanlen, C.-E. Cella: "Deep Convolutional Networks on the Pitch Spiral for Music Instrument Recognition", *Proc. ISMIR*, 2016.

[19] M.A. Loureiro, H. Bastos de Paula and H.C. Yehia: "Timbre Classification Of A Single Musical Instrument". *Proc. ISMIR*, 2004.

[20] B. McFee, C. Raffel, D. Liang, D.P.W. Ellis, M. McVicar, E. Battenberg, and O. Nieto: "Librosa: Audio and Music Signal Analysis in Python", *Proc. of the 14th Python in Science Conf. (SCIPY)*, 2015.

[21] K. Patil, M. Elhilali, "Biomimetic Spectro-Temporal Features for Music Instrument Recognition in Isolated Notes and Solo phrases", *EURASIP J. Audio Speech Music Process*, 2015.

[22] C. Perret: "Une Rencontre entre Musique Savante et Jazz, Musique de Tradition Orale et les œuvres aux Accents Jazzistiques d'Érik Satie, Darius Milhaud, Igor Stravinsky et Maurice Ravel", *Volume !*, Vol. 2:1, pp. 43-67, 2003.

[23] J. Pons, O. Slizovskaia, R. Gong, E. Gómez and X. Serra, "Timbre Analysis of Music Audio Signals with Convolutional Neural Networks", *25th European Signal Processing Conference (EUSIPCO)*, 2017.

[24] R. Rowe: *Interactive Music Systems: Machine Listening and Composing*, Chapter 5, MIT Press, Cambridge MA, USA, 1993.

[25] K. Saito, K. Watanabe, Y. Ushiku, T. Harada: "Maximum Classifier Discrepancy for Unsupervised Domain Adaptation", arXiv:1712.02560, 2018.

[26] P. Schaeffer: *Traité des objets musicaux*, Editions du Seuil, pp. 52-53, 1966.

[27] A.R. Tindale, A. Kapur, G. Tzanetakis, and I. Fujinaga: "Retrieval of Percussion Gestures using Timbre Classification Techniques", *Proc. ISMIR*, 2004.

[28] J. Wiederker: *The contemporary Cello*, pp. 59-61, Ed. L'Oiseau d'or, Sainte-Geneviève-des-Bois, France.