LEARNING PITCH INVARIANTS FOR INSTRUMENT RECOGNITION

Names should be omitted for double-blind reviewing
Affiliations should be omitted for double-blind reviewing

ABSTRACT

Musical performance combines a wide range of pitches, nuances, and expressive techniques. Audio-based classification of musical instruments thus requires to build signal representations that are invariant to such transformations. This article investigates the construction of multi-stage architectures for instrument recognition, given a limited amount of annotated training data. We show that the main drawback of mel-frequency cepstral coefficients (MFCC) resides in their lack of invariance with respect to realistic pitch shifts, despite being designed to be invariant to the frequency transposition of pure tones. In turn, a deep convolutional network (ConvNet) in the time-frequency domain is able to disentangle pitch from timbral information, hence yielding better classification accuracy. We further improve the learned representation by introducing novel weight sharing strategies dedicated to quasi-harmonic sounds with fixed spectral envelope, which are archetypal of musical notes.

1. INTRODUCTION

Among the cognitive attributes of musical tones, pitch is distinguished by a combination of three properties. First, it is relative: ordering pitches from low to high gives rise to intervals and melodic patterns. Secondly, it is intensive: multiple pitches heard simultaneously produce a chord, not a single unified tone – contrary to loudness, which adds up with the number of sources. Thirdly, it does not depend on instrumentation: this makes possible the transcription of polyphonic music under a single symbolic system [5].

Tuning auditory filters to a perceptual scale of pitches provides a time-frequency representation of music signals that satisfies the first two of these properties. It is thus a starting point for a wide range of MIR applications, which can be separated in two categories: pitch-*relative* (e.g. chord estimation [14]) and pitch-*invariant* (e.g. instrument recognition [9]). Both aim at disentangling pitch from timbral content as independent factors of variability, a goal that is made possible by the third aforementioned property. This is pursued by extracting mid-level features on top of the spectrogram, be them engineered or learned from training data. Both approaches have their limitations: a "bag-of-features" lacks flexibility to represent fine-grain class boundaries, whereas a purely learned pipeline often leads to uninterpretable overfitting, especially in MIR where the quantity of thoroughly annotated data is relatively small.

In this article, we strive to integrate domain-specific knowledge about musical pitch into a deep learning framework, in an effort towards bridging the gap between feature engineering and feature learning.

Section 2 reviews the related work on feature learning for signal-based music classification. Section 3 demonstrates that pitch is the major factor of variability among musical notes of a given instrument, if described by their mel-frequency cepstra. Section 4 presents a typical deep learning architecture for spectrogram-based classification, consisting of two convolutional layers in the time-frequency domain and one densely connected layer. Section 5 introduces alternative convolutional architectures for learning mid-level features, along time and along a Shepard pitch spiral, as well as aggregation of multiple models in the deepest layers. Sections 6 discusses the effectiveness of the presented systems on a challenging dataset for music instrument recognition.

2. RELATED WORK

Spurred by the growth of annotated datasets and the democratization of high-performance computing, feature learning has enjoyed a renewed interest in recent years within the MIR community, both in supervised and unsupervised settings. Whereas unsupervised learning (e.g. k-means [26], Gaussian mixtures [15]) is employed to fit the distribution of the data with few parameters of relatively low abstraction and high dimensionality, state-of-the-art supervised learning consists of a deep composition of multiple nonlinear transformations, jointly optimized to predict class labels, and whose behaviour tend to gain in abstraction as depth increases [28].

As compared to other deep learning techniques for audio processing, convolutional networks happen to strike the balance between learning capacity and robustness. The convolutional structure of learned transformations is derived from the assumption that the input signal, be it a one-dimensional waveform or a two-dimensional spectrogram, is stationary — which means that content is independent from location. Moreover, the most informative dependencies between signal coefficients are assumed to be concentrated to temporal or spectrotemporal neighborhoods. Under such hypotheses, linear transformations can be learned efficiently by limiting their support to a small kernel which is convolved over the whole input. This method, known as weight sharing, decreases the number of parameters of each feature map while increasing the amount of data on which kernels are trained.

By design, convolutional networks seem well adapted to instrument recognition, as this task does not require a precise timing of the activation function, and is thus essentially a challenge of temporal integration [9, 15]. Furthermore, it benefits from an unequivocal ground truth, and may be simplified to a single-label classification problem by extracting individual stems from a multitrack dataset [2]. As such, it is often used a test bed for the development of new algorithms [18, 19], as well as in computational studies in music cognition [21, 22].

Some other applications of deep convolutional networks include onset detection [24], transcription [25], chord recognition [14], genre classification [3], downbeat tracking [8], boundary detection [27], and recommendation [28].

Interestingly, many research teams in MIR have converged to employ the same architecture, consisting of two convolutional layers and two densely connected layers [7,14,16,18,19,24,27], and this article makes no exception. However, there is no clear consensus regarding the weight sharing strategies that should be applied to musical audio streams: convolutions in time or in time-frequency coexist in the recent literature. A promising paradigm [6, 8], at the interaction between feature engineering and feature learning, is to extract temporal or spectrotemporal descriptors of various low-level modalities, train specific convolutional layers on each modality to learn mid-level features, and hybridize information at the top level. Recognizing that this idea has been successfully applied to large-scale artist recognition [6] as well as downbeat tracking [8], we aim to proceed in a comparable way for instrument recognition.

3. HOW INVARIANT IS THE MEL-FREQUENCY CEPSTRUM ?

The mel scale is a quasi-logarithmic function of acoustic frequency designed such that perceptually similar pitch intervals appear equal in width over the full hearing range. This section shows that engineering transposition-invariant features from the mel scale does not suffice to build pitch invariants for complex sounds, thus motivating further inquiry.

The time-frequency domain produced by a constant-Q filter bank tuned to the mel scale is covariant with respect to pitch transposition of pure tones. As a result, a chromatic scale played at constant speed would draw parallel, diagonal lines, each of them corresponding to a different partial wave. However, the physics of musical instruments constrain these partial waves to bear a negligible energy if their frequencies are beyond the range of acoustic resonance.

As shown on Figure 1, the constant-Q spectrogram of a tuba chromatic scale exhibits a fixed, cutoff frequency at about 2.5 kHz, which delineates the support of its spectral envelope. This elementary observation implies that re-

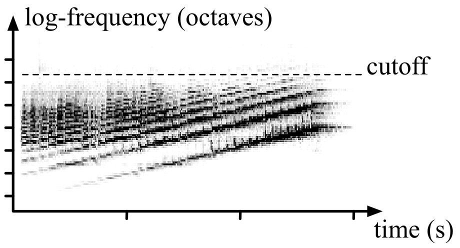

Figure 1: Constant-Q spectrogram of a chromatic scale played by a tuba. Although the harmonic partials shift progressively, the spectral envelope remains unchanged, as revealed by the presence of a fixed cutoff frequency. See text for details.

alistic pitch changes cannot be modeled by translating a rigid spectral template along the log-frequency axis. The same property is verified for a wide class of instruments, especially brass and woodwinds. As a consequence, the construction of powerful invariants to musical pitch is not amenable to delocalized operations on the mel-frequency spectrum, such as a discrete cosine transform (DCT) which leads to the mel-frequency cepstral coefficients (MFCC) classically used in music classification [9, 15].

To validate the above claim, we have extracted the MFCC of 1116 individual notes from the RWC dataset [10], as played by 6 instruments, with 32 pitches, 3 nuances, and 2 interprets and manufacturers. When more than 32 pitches were available (e.g. piano), we selected a contiguous subset of 32 pitches in the middle register. Following a well-established rule [9, 15], the MFCC were defined the 12 lowest nonzero "quefrencies" among the DCT coefficients extracted from a filter bank of 40 mel-frequency bands. We then have computed the distribution of squared Euclidean distances between musical notes in the 12-dimensional space of MFCC features.

Figure 2 summarizes our results. We found that restricting the cluster to one nuance, one interpret, or one manufacturer hardly reduces intra-class distances. This suggests that MFCC are fairly successful in building invariant representations to such factors of variability. In contrast, the cluster corresponding to each instrument is shrinked if decomposed into a mixture of same-pitch clusters, sometimes by an order of magnitude. In other words, most of the variance in an instrument cluster of mel-frequency cepstra is due to pitch transposition.

Keeping less than 12 coefficients certainly improves invariance, yet at the cost of inter-class discriminability, and vice versa. This experiment shows that the mel-frequency cepstrum is perfectible in terms of invariance-discriminability tradeoff, and that there remains a lot to be gained by feature learning in this area.

4. DEEP CONVOLUTIONAL NETWORKS

A deep learning system for classification is built by stacking multiple layers of weakly nonlinear transformations, whose parameters are optimized such that the top-level

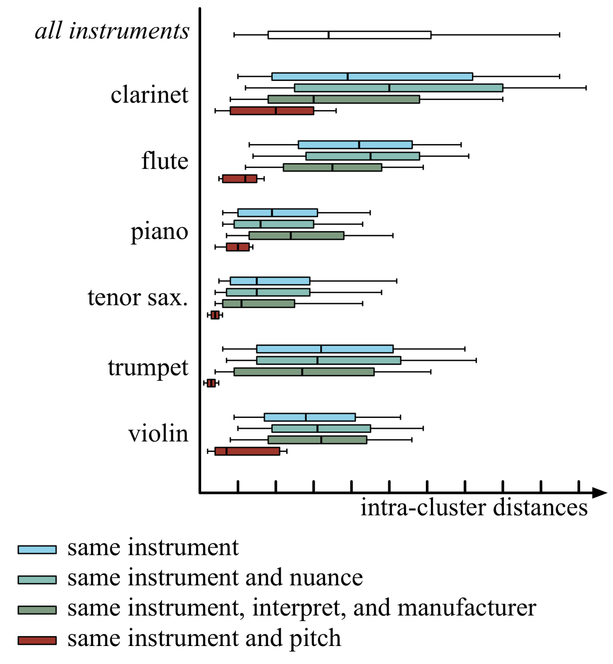

Figure 2: Distributions of squared Euclidean distances among various MFCC clusters in the RWC dataset. Whisker ends denote lower and upper deciles. See text for details.

layer fits a training set of labeled examples. This section introduces a typical deep learning architecture for audio classification and describes the functioning of each layer.

Each layer in a convolutional network typically consists in the composition of three operations: two-dimensional convolutions, application of a pointwise nonlinearity, and local pooling. The deep feed-forward network made of two convolutional layers and two densely connected layers, on which our experiment are conducted, has become a *de facto* standard in the MIR community [7, 14, 16, 18, 19, 24, 27]. This ubiquity in the literature suggests that the a four-layer network with two convolutional layers is well adapted to supervised audio classification problems of moderate size.

The input of our system is a constant-Q spectrogram, which is very comparable to a mel-frequency spectrogram. We used the implementation from the librosa package [20] with $Q = 12$ filters per octave, center frequencies ranging from A_1 (55 Hz) to A_9 (14 kHz), and a hop size of 23 ms. Furthermore, we applied nonlinear perceptual weighting of loudness in order to reduce the dynamic range between the fundamental partial and its upper harmonics. A 3-second sound excerpt $\boldsymbol{x}[t]$ is represented by a time-frequency matrix $\boldsymbol{x_1}[t, k_1]$ of width $T = 128$ samples and height $K_1 = 96$ frequency bands.

A convolutional operator is defined as a family $\boldsymbol{W_2}[\tau, \kappa_1, k_2]$ of K_2 two-dimensional filters, whose impulse repsonses are all constrained to have width Δt and height Δk_1. Element-wise biases $\boldsymbol{b_2}[k_2]$ are added to the convolutions, resulting in the three-way tensor

$$\begin{aligned}&\boldsymbol{y_2}[t, k_1, k_2]\\ &= \boldsymbol{b_2}[k_2] + \boldsymbol{W_2}[t, k_1, k_2] \overset{t,k_1}{*} \boldsymbol{x_1}[t, k_1]\\ &= \boldsymbol{b_2}[k_2] + \sum_{\substack{0\leq\tau<\Delta t\\ 0\leq\kappa_1<\Delta k_1}} \boldsymbol{W_2}[\tau, \kappa_1, k_2]\boldsymbol{x_1}[t-\tau, k_1-\kappa_1]. \quad (1)\end{aligned}$$

The pointwise nonlinearity we have chosen is the rectified linear unit (ReLU), with a rectifying slope of $\alpha = 0.3$ for negative inputs.

$$\boldsymbol{y_2^+}[t, k_1, k_2] = \begin{cases}\alpha\boldsymbol{x_2}[t, k_1, k_2] & \text{if } \boldsymbol{x_2}[t, k_1, k_2] < 0\\ \boldsymbol{x_2}[t, k_1, k_2] & \text{if } \boldsymbol{x_2}[t, k_1, k_2] > 0\end{cases} \quad (2)$$

The pooling step consists in retaining the maximal activation among neighboring units in the time-frequency domain (t, k_1) over non-overlapping rectangles of width Δt and height Δk_1.

$$\boldsymbol{x_2}[t, k_1, k_2] = \max_{\substack{0\leq\tau<\Delta t\\ 0\leq\kappa_1<\Delta k_1}} \left\{\boldsymbol{y_2^+}[t-\tau, k_1-\kappa_1, k_2]\right\} \quad (3)$$

The hidden units in $\boldsymbol{x_2}$ are in turn fed to a second layer of convolutions, ReLU, and pooling. Observe that the corresponding convolutional operator $\boldsymbol{W_3}[\tau, \kappa_1, k_2, k_3]$ performs a linear combination of time-frequency feature maps in $\boldsymbol{x_2}$ along the channel variable k_2.

$$\begin{aligned}&\boldsymbol{y_3}[t, k_1, k_3]\\ &= \sum_{k_2} \boldsymbol{b_3}[k_2, k_3] + \boldsymbol{W_3}[t, k_1, k_2, k_3] \overset{t,k_1}{*} \boldsymbol{x_2}[t, k_1, k_2]. \quad (4)\end{aligned}$$

Tensors $\boldsymbol{y_3^+}$ and $\boldsymbol{x_3}$ are derived from $\boldsymbol{y_3}$ by ReLU and pooling, with formulae similar to Eqs. (2) and (3). The third layer consists of the linear projection of $\boldsymbol{x_3}$, viewed as a vector of the flattened index (t, k_1, k_3), over K_4 units:

$$\boldsymbol{y_4}[k_4] = \boldsymbol{b_4}[k_4] + \sum_{t,k_1,k_3} \boldsymbol{W_4}[t, k_1, k_3, k_4]\boldsymbol{x_3}[t, k_1, k_3] \quad (5)$$

We apply a ReLU to $\boldsymbol{y_4}$, yielding $\boldsymbol{x_4}[k_4] = \boldsymbol{y_4^+}[k_4]$. Finally, we project $\boldsymbol{x_4}$ onto a layer of output units $\boldsymbol{y_5}$ that should represent instrument activations: $\boldsymbol{y_5}[k_5] = \sum_{k_4} \boldsymbol{W_5}[k_4, k_5]\boldsymbol{x_4}[k_4]$. The final transformation is a softmax nonlinearity, which ensures that output coefficients are non-negative and sum to one, hence can be fit to a probability distribution:

$$\boldsymbol{x_5}[k_5] = \frac{\exp \boldsymbol{y_5}[k_5]}{\sum_{\kappa_5} \exp \boldsymbol{y_5}[\kappa_5]}. \quad (6)$$

Given a training set of spectrogram-instrument pairs $(\boldsymbol{x_1}, k)$, all weigths in the network are iteratively updated to minimize the cross-entropy loss $\mathscr{L}(\boldsymbol{x_5}, k) = -\log \boldsymbol{x_5}[k]$ over shuffled mini-batches of size 32 with uniform class distribution. The pairs $(\boldsymbol{x_1}, k)$ are extracted on the fly by selecting non-silent regions at random within a dataset of single-instrument audio recordings. Each 3-second spectrogram $\boldsymbol{x_1}[t, k_1]$ within a batch is globally normalized such that the whole batch had zero mean and unit

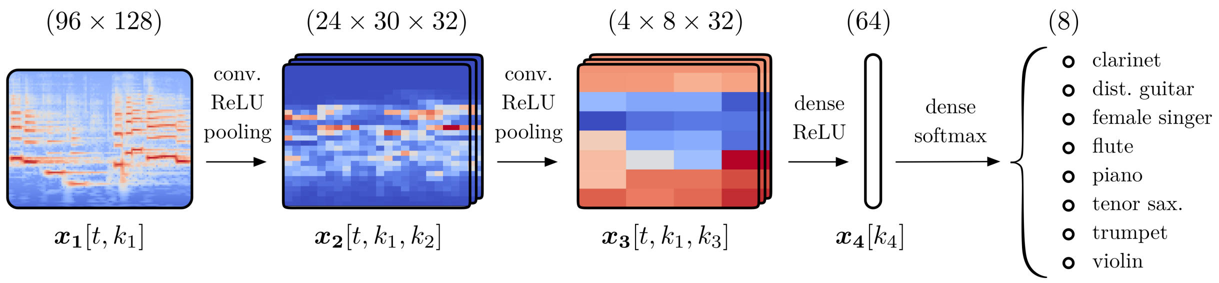

Figure 3: A two-dimensional deep convolutional network trained on constant-Q spectrograms. See text for details.

variance. At training time, a random dropout of 50% is applied to the activations of $\boldsymbol{x_3}$ and $\boldsymbol{x_4}$. The learning rate policy for each scalar weight in the network is Adam [17], a state-of-the-art online optimizer for gradient-based learning. Mini-batch training is stopped after the average training loss stopped decreasing over one full epoch of size 8192. The architecture is built using the Keras library [4] and trained on a graphics processing unit within a few minutes.

5. IMPROVED WEIGHT SHARING STRATEGIES

Although a dataset of music signals is unquestionably stationary over the time dimension – at least at the scale of a few seconds – it cannot be taken for granted that all frequency bands of a constant-Q spectrogram would have the same local statistics [13]. In this section, we introduce two alternative architectures to address the nonstationarity of music on the log-frequency axis, while still leveraging the efficiency of convolutional representations in the time-frequency domain.

Many are the objections to the stationarity assumption among local neighborhoods in mel frequency. Notably enough, one of the most compelling is derived from the classical source-filter model of sound production. The filter, which carries the overall spectral envelope, is affected by intensity and playing style, but not by pitch. Conversely, the source, which consists of a pseudo-periodic wave, is transposed in frequency under the action of pitch. In order to extract the discriminative information present in both terms, it is first necessary to disentangle the contributions of source and filter in the constant-Q spectrogram. Yet, this can only be achieved by exploiting long-range correlations in frequency, such as harmonic and formantic structures. Besides, the harmonic comb created by the Fourier series of the source makes an irregular pattern on the log-frequency axis, which is hard to characterize by local statistics.

5.1 One-dimensional convolutions at high frequencies

Facing nonstationary constant-Q spectra, the most conservative workaround is to increase the height $\Delta\kappa_1$ of each convolutional kernel up to the total number of bins K_1 in the spectrogram. As a result, $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ are no longer transposed over adjacent frequency bands, since convolutions are merely performed over the time variable. The definition of $\boldsymbol{y_2}[t, k_1, k_2]$ rewrites as

$$\begin{aligned}
&\boldsymbol{y_2}[t, k_1, k_2] \\
&= \boldsymbol{b_2}[k_2] + \boldsymbol{W_2}[t, k_1, k_2] \overset{t}{*} \boldsymbol{x_1}[t, k_1] \\
&= \boldsymbol{b_2}[k_2] + \sum_{0 \leq \tau < \Delta t} \boldsymbol{W_2}[\tau, k_1, k_2] \boldsymbol{x_1}[t-\tau, k_1], \qquad (7)
\end{aligned}$$

and similarly for $\boldsymbol{y_3}[t, k_1, k_3]$. While this approach is theoretically capable of encoding pitch invariants, it is prone to early specialization of low-level features, thus not fully taking advantage of the network depth.

However, the situation is improved if the one-dimensional kernels are restricted to the highest frequencies in the constant-Q spectrum. It should be observed that, around the n^{th} partial of a quasi-harmonic sound, the distance in log-frequency between neighboring partials decays like $1/n$, and the unevenness between those distances decays like $1/n^2$. Consequently, at the topmost octaves of the constant-Q spectrum, where n is equal or greater than Q, the partials appear close to each other and almost evenly spaced. Furthermore, due to the logarithmic compression of loudness, the polynomial decay of the spectral envelope is linearized: thus, at high frequencies, transposed pitches have similar spectra up to some additive bias. The combination of these two phenomena implies that the correlation between constant-Q spectra of different pitches is greater towards high frequencies, and that the learning of polyvalent feature maps becomes tractable.

In our experiments, the one-dimensional convolutions over the time variable range from A_6 $(1,76\,\text{kHz})$ to A_9 $(14\,\text{kHz})$.

5.2 Convolutions on the pitch spiral at low frequencies

The weight sharing strategy presented above exploits the facts that, at high frequencies, quasi-harmonic partials are numerous, and that the amount of energy within a frequency band is independent of pitch. At low frequencies, we make the exact opposite assumptions: we claim that the harmonic comb is sparse and covariant with respect to pitch shift. Observe that, for any two distinct partials taken at random between 1 and n, the probability that they are in octave relation is slightly above $1/n$. Thus, for n relatively low, the structure of harmonic sounds is well described by merely measuring correlations between partials one octave apart. This idea consists in rolling up the log-frequency axis into a Shepard pitch spiral, such that octave intervals correspond to full turns, hence aligning all coefficients of the form $\boldsymbol{x_1}[t, k_1 + Q \times j_1]$ for $j_1 \in \mathbb{Z}$ onto the same radius of the spiral. Henceforth, correlations between power-of-two harmonics are revealed by the octave variable j_1.

To implement a convolutional network on the pitch spiral, we crop the constant-Q spectrogram in log-frequency into $J_1 = 3$ half-overlapping bands whose height equals $2Q$, that is two octaves. Each feature map in the first layer, indexed by k_2, results from the sum of convolutions between a time-frequency kernel and a band, thus emulating a linear combination in the pitch spiral with a 3-d tensor $\boldsymbol{W_2}[\tau, \kappa_1, j_1, k_2]$ at fixed k_2. The definition of $\boldsymbol{y_2}[t, k_1, k_2]$ rewrites as

$$\begin{aligned}\boldsymbol{y_2}[t, k_1, k_2] = \boldsymbol{b_2}[k_2] \\ + \sum_{\tau, \kappa_1, j_1} \boldsymbol{W_2}[\tau, \kappa_1, j_1, k_2] \\ \times \boldsymbol{x_1}[t - \tau, k_1 - \kappa_1 - Qj_1]. \quad (8)\end{aligned}$$

The above is different from training two-dimensional kernel on a time-chroma-octave tensor, since it does not suffer from artifacts at octave boundaries.

The linear combinations of frequency bands that are one octave apart, as proposed here, bears a resemblance with engineered features for music instrument recognition [23], such as tristimulus, empirical inharmonicity, harmonic spectral deviation, odd-to-even harmonic energy ratio, as well as octave band signal intensities (OBSI) [15].

Guaranteeing the partial index n to remain low is achieved by restricting the pitch spiral to its lowest frequencies. This operation also partially circumvents the problem of fixed spectral envelope in musical sounds, thus improving the validness of the stationarity assumption. In our experiments, the pitch spiral ranges from A_2 ($110\,\mathrm{Hz}$) to A_6 ($1,76\,\mathrm{kHz}$).

In summary, the classical two-dimensional convolutions make a stationarity assumption among frequency neighborhoods. This approach gives a coarse approximation of the spectral envelope. Resorting to one-dimensional convolutions allows to disregard nonstationarity, but does not yield a pitch-invariant representation per se: thus, we only apply them at the topmost frequencies, i.e. where the invariance-to-stationarity ratio in the data is already favorable. Conversely, two-dimensional convolutions on the pitch spiral addresses the invariant representation of sparse, transposition-covariant spectra: as such, they are best suited to the lowest frequencies, i.e. where partials are further apart and pitch changes can be approximated by log-frequency translations. The next section reports experiments on instrument recognition that capitalize on these considerations.

	minutes	tracks	minutes	tracks
clarinet	10	7	13	18
dist. guitar	15	14	17	11
female singer	10	11	19	12
flute	7	5	53	29
piano	58	28	44	15
tenor sax.	3	3	6	5
trumpet	4	6	7	27
violin	51	14	49	22
total	158	88	208	139

Table 1: Quantity of data in the training set (left) and test set (right). The training set is derived from MedleyDB. The test set is derived from MedleyDB for distorted electric guitar and female singer, and from [15] for other instruments.

6. APPLICATIONS

The proposed algorithms are trained on a subset of MedleyDB v1.1. [2], a dataset of 122 multitracks annotated with instrument activations. We extracted the monophonic stems corresponding to a selection of eight pitched instruments (see Table 1). Stems with leaking instruments in the background were discarded.

The evaluation set consists of 126 recordings of solo music collected by Joder et al. [15], supplemented with 23 stems of electric guitar and female voice from MedleyDB. In doing so, guitarists and vocalists were thoroughly put either in the training set or the test set, to prevent any artist bias. We discarded recordings with extended instrumental techniques, since they are extremely rare in MedleyDB. Constant-Q spectrograms from the evaluation set were split into half-overlapping, 3-second excerpts.

For the two-dimensional convolutional network, each of the two layers consists of 32 kernels of width 5 and height 5, followed by a max-pooling of width 5 and height 3. Expressed in physical units, the supports of the kernels are respectively equal to $116\,\mathrm{ms}$ and $580\,\mathrm{ms}$ in time, 5 and 10 semitones in frequency. For the one-dimensional convolutional network, each of two layers consists of 32 kernels of width 3, followed by a max-pooling of width 5. Observe that the temporal supports match those of the two-dimensional convolutional network. For the convolutional network on the pitch spiral, the first layer consists of 32 kernels of width 5, height 3 semitones, and a radial length of 3 octaves in the spiral. The max-pooling operator and the second layer are the same as in the two-dimensional convolutional network.

In addition to the three architectures above, we build hybrid networks implementing more than one of the weight sharing strategy presented above. In all architectures, the densely connected layers have $K_4 = 64$ hidden units and $K_5 = 8$ output units.

In order to compare the results against shallow classifiers, we also extracted a typical "bag-of-features" over half-overlapping, 3-second excerpts in the training set. These features consist of the means and standard devia-

	clarinet	distorted	female	flute	piano	tenor	trumpet	violin	average
bag-of-features	49.9	92.7	81.6	22.5	**99.7**	4.4	63.7	**76.2**	61.4
and random forest	(± 0.8)	(± 0.4)	(± 1.5)	(± 0.8)	**(± 0.1)**	(± 1.1)	(± 2.1)	(± 3.1)	(± 0.5)
1-d	28.8	**91.8**	82.9	**51.3**	73.3	**59.0**	63.3	43.9	61.8
(20k parameters)	(± 5.0)	**(± 1.1)**	(± 1.9)	**(± 13.4)**	(± 11.0)	**(± 6.8)**	(± 5.0)	(± 6.3)	(± 0.9)
spiral	61.1	72.3	84.4	30.0	86.9	52.7	54.9	37.0	59.9
(36k parameters)	(± 8.7)	(± 6.2)	(± 6.1)	(± 4.0)	(± 5.8)	(± 16.4)	(± 6.6)	(± 5.6)	(± 2.4)
2-d, 32 kernels	81.3	86.0	80.6	44.4	96.8	48.4	68.0	68.5	69.1
(93k parameters)	(± 4.1)	(± 2.7)	(± 1.7)	(± 4.4)	(± 1.4)	(± 5.3)	(± 6.2)	(± 9.3)	(± 2.0)
spiral & 1-d	79.6	90.2	84.5	41.8	96.5	53.0	59.8	47.6	69.1
(55k parameters)	(± 2.1)	(± 2.3)	(± 2.8)	(± 4.1)	(± 2.3)	(± 16.5)	(± 1.9)	(± 6.1)	(± 2.0)
2-d & 1-d	73.3	86.3	81.0	49.5	96.5	55.0	67.7	72.4	73.8
(111k parameters)	(± 6.4)	(± 5.2)	(± 5.5)	(± 6.9)	(± 0.9)	(± 11.5)	(± 2.5)	(± 5.9)	(± 2.3)
2-d & spiral	**82.3**	86.5	**86.9**	45.8	97.6	51.2	66.9	73.3	71.7
(128k parameters)	**(± 3.2)**	(± 4.5)	**(± 3.6)**	(± 2.9)	(± 0.8)	(± 10.6)	(± 5.8)	(± 4.4)	(± 2.0)
2-d & 1-d & spiral	75.0	88.0	85.9	48.3	97.8	**59.0**	67.3	70.9	**74.0**
(147k parameters)	(± 4.3)	(± 3.7)	(± 3.8)	(± 6.6)	(± 0.6)	**(± 7.3)**	(± 4.4)	(± 6.1)	**(± 0.6)**
2-d, 48 kernels	77.4	84.5	84.2	45.5	96.5	52.6	**68.8**	69.3	71.7
(158k parameters)	(± 6.0)	(± 2.5)	(± 5.7)	(± 7.3)	(± 1.4)	(± 10.1)	(± 1.8)	(± 7.2)	(± 2.0)

Table 2: Test set accuracies for all presented architectures. All convolutional layers have 32 kernels unless stated otherwise.

tions of spectral shape descriptors, i.e. centroid, bandwidth, skewness, and rolloff ; the mean and standard deviation of the zero-crossing rate in the time domain ; the mean MFCC as well their first and second derivative. We trained a random forest of 100 decision trees on the resulting feature vector of dimension 70, with uniform class probability.

Results are summarized in Table 2. First of all, the bag-of-features approach presents large accuracy variations between classes, due to the unbalance of available training data. In contrast, most convolutional models, especially hybrid ones, show less correlation between the amount of training data in the class and the accuracy. This suggests that convolutional networks are able to learn polyvalent mid-level features that can be re-used a test time to discriminate rarer classes.

Furthermore, 2-d convolutions outperform other non-hybrid weight sharing strategies. However, a class with broadband temporal modulations, namely the distorted electric guitar, is best classified with 1-d convolutions.

Hybridizing 2-d with either 1-d or spiral convolutions provide consistent, albeit small improvements with respect to 2-d alone. The best overall accuracy is reached by the full hybridization of all three weight sharing strategies, because of a performance boost for the rarest classes.

The accuracy gain by combining multiple models could simply be the result of a greater number of parameters. To refute this hypothesis, we train a 2-d convolutional network with 48 kernels instead of 32, so as to match the budget of the full hybrid model, i.e. about 150k parameters. The performance is certainly increased, but not up to the hybrid models involving 2-d convolutions, which have less parameters. Increasing the number of kernels even more cause the accuracy to level out and the variance between trials to increase.

Running the same experiments with broader frequency ranges of 1-d and spiral convolutions often lead to a degraded performance, and are thus not reported.

7. CONCLUSIONS

Understanding the influence of pitch in audio streams is paramount to the design of an efficient system for automated classification, tagging, and similarity retrieval in music. We have presented deep learning methods to address pitch invariance while preserving good timbral discriminability. It consists in training a feed-forward convolutional network over the constant-Q spectrogram, with three different weight sharing strategies according to the type of input: along time at high frequencies (above 2 kHz), on a pitch spiral at low frequencies (below 2 kHz), and in time-frequency over both high and low frequencies.

A possible improvement of the presented architecture would be to place a third convolutional layer in the time domain before performing long-term max-pooling, hence modelling the joint dynamics of the three mid-level feature maps. Future work will investigate the association of the presented weight sharing strategies with recent advances in deep learning for music informatics, such as data augmentation [19], multiscale representations [1,11], and adversarial training [16].

8. REFERENCES

[1] Joakim Andén, Vincent Lostanlen, and Stéphane Mallat. Joint time-frequency scattering for audio classification. In *Proc. MLSP*, 2015.

[2] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. Med-

leyDB: a multitrack dataset for annotation-intensive MIR research. In *Proc. ISMIR*, 2014.

[3] Keunwoo Choi, George Fazekas, Mark Sandler, Jeonghee Kim, and Naver Labs. Auralisation of deep convolutional neural networks: listening to learned features. In *Proc. ISMIR*, 2015.

[4] François Chollet. Keras: a deep learning library for Theano and TensorFlow, 2015.

[5] Alain de Cheveigné. Pitch perception. In *Oxford Handbook of Auditory Science: Hearing*, chapter 4, pages 71–104. Oxford University Press, 2005.

[6] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *Proc. ISMIR*, 2011.

[7] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Proc. ICASSP*, 2014.

[8] Simon Durand, Juan P. Bello, Bertrand David, and Gaël Richard. Feature-adapted convolutional neural networks for downbeat tracking. In *Proc. ICASSP*, 2016.

[9] Antti Eronen and Anssi Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *Proc. ICASSP*, 2000.

[10] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: music genre database and musical instrument sound database. In *Proc. ISMIR*, 2003.

[11] Philippe Hamel, Yoshua Bengio, and Douglas Eck. Building musically-relevant audio features through multiple timescale representations. In *Proc. ISMIR*, 2012.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. ICCV*, 2015.

[13] Eric J. Humphrey, Juan P. Bello, and Yann Le Cun. Feature learning and deep architectures: New directions for music informatics. *JIIS*, 41(3):461–481, 2013.

[14] Eric J. Humphrey, Taemin Cho, and Juan P. Bello. Learning a robust tonnetz-space transform for automatic chord recognition. In *Proc. ICASSP*, 2012.

[15] Cyril Joder, Slim Essid, and Gaël Richard. Temporal integration for audio classification with application to musical instrument classification. *IEEE TASLP*, 17(1):174–186, 2009.

[16] Corey Kereliuk, Bob L. Sturm, and Jan Larsen. Deep Learning and Music Adversaries. *IEEE Trans. Multimedia*, 17(11):2059–2071, 2015.

[17] Diederik P. Kingma and Jimmy Lei Ba. Adam: a method for stochastic optimization. In *Proc. ICML*, 2015.

[18] Peter Li, Jiyuan Qian, and Tian Wang. Automatic instrument recognition in polyphonic music using convolutional neural networks. *arXiv preprint*, 1511.05520, 2015.

[19] Brian McFee, Eric J. Humphrey, and Juan P. Bello. A software framework for musical data augmentation. In *Proc. ISMIR*, 2015.

[20] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, Oriol Nieto, Eric Battenberg, Josh Moore, Dan Ellis, Ryuichi Yamamoto, Rachel Bittner, Douglas Repetto, Petr Viktorin, João Felipe Santos, and Adrian Holovaty. librosa: 0.4.1. zenodo. 10.5281/zenodo.18369, October 2015.

[21] Michael J. Newton and Leslie S. Smith. A neurally inspired musical instrument classification system based upon the sound onset. *JASA*, 131(6):4785, 2012.

[22] Kailash Patil, Daniel Pressnitzer, Shihab Shamma, and Mounya Elhilali. Music in our ears: the biological bases of musical timbre perception. *PLoS Comput. Biol.*, 8(11):e1002759, 2012.

[23] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, Ircam, 2004.

[24] Jan Schlüter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *Proc. ICASSP*, 2014.

[25] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An end-to-end neural network for polyphonic music transcription. *arXiv preprint*, page 1508.01774, 2015.

[26] Dan Stowell and Mark D. Plumbley. Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. *PeerJ*, 2:e488, 2014.

[27] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary detection in music structure analysis using convolutional neural networks. In *Proc. ISMIR*, 2014.

[28] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Proc. NIPS*, 2013.