Experimenting with Musically Motivated Convolutional Neural Networks

Jordi Pons
Music Technology Group,
Universitat Pompeu Fabra,
Barcelona
jordi.pons@upf.edu

Thomas Lidy
Institute of Software Technology
and Interactive Systems,
TU Wien
lidy@ifs.tuwien.ac.at

Xavier Serra
Music Technology Group,
Universitat Pompeu Fabra,
Barcelona
xavier.serra@upf.edu

***Abstract*—A common criticism of deep learning relates to the difficulty in understanding the underlying relationships that the neural networks are learning, thus behaving like a black-box. In this article we explore various architectural choices of relevance for music signals classification tasks in order to start understanding what the chosen networks are learning. We first discuss how convolutional filters with different shapes can fit specific musical concepts and based on that we propose several musically motivated architectures. These architectures are then assessed by measuring the accuracy of the deep learning model in the prediction of various music classes using a known dataset of audio recordings of ballroom music. The classes in this dataset have a strong correlation with tempo, what allows assessing if the proposed architectures are learning frequency and/or time dependencies. Additionally, a black-box model is proposed as a baseline for comparison. With these experiments we have been able to understand what some deep learning based algorithms can learn from a particular set of data.**

I. Introduction

The amount of available music audio recordings is constantly growing. However, most of the recordings are poorly labeled and this difficult its identification and access. Indexing such musical content with semantic labels has been a research topic within the field of music information research (MIR) for the past two decades. Having such semantic information per music track would allow to better organize the existing music repositories and would enable users to better explore the music collection space, what would increase the retrieval and use possibilities.

Nowadays, deep learning approaches irrupted strongly into the MIR community. Even some researchers declare that is the time for a paradigm shift: from hand-crafted features and shallow classifiers to deep processing models [8]. A brief review of the state-of-the-art in MIR and deep learning reveals that such algorithms achieved competitive results in a relatively short amount of time – most relevant papers were published during the last 5 years. Many researchers successfully used deep learning for several tasks: onset detection [17], genre classification [2], chord estimation [23], auto-tagging [4] or source separation [7]. However, not only good results are supporting the strong irruption of this technology in the MIR field, deep learning underlying conceptual construction can be advantageous for musical analysis:

- Music is hierarchic in frequency (note, chord) and time (onset, rhythm). Deep learning can naturally allow this hierarchic representation since its architecture is inherently hierarchical due to its depth.
- Relationships between musical events in the time domain are important for human music perception. Using recurrent neural networks [5] (RNNs) and/or convolutional neural networks [9] (CNNs), the net is capable to analyze such temporal context. RNNs can model long-term dependencies (music structure or recurrent harmonies) and CNNs can model the local context (instrument's timbre or musical units). RNNs can also model short-term dependencies, meaning that by architectural choices researchers can tailor the net towards learning musical aspects in manifold ways.

Regardless of the competitive results achieved and the conceptual benefits of using deep learning approaches, there is still a lack of understanding. We still do not fully grasp what the nets are learning. Dieleman *et al.* made some progress showing that "higher-level features are defined in terms of lower-level features" for music [3]. They found[1] that the first convolutional layer in their deep learning music recommendation system had filters specialized in low-level musical concepts (vibrato, vocal thirds, pitches, chords), whereas the third convolutional layer filters were specialized in higher-level musical concepts (christian rock, chinese pop, 8-bit). This matches with similar results found by the image processing research community where lower layers are capable of learning shapes that are combined in higher layers to represent objects [22]. Furthermore, Dieleman *et al.* [2] also proposed a deep learning algorithm that preserves musically significant timescales (beats-bars-themes) within the design of the architecture, what "leads to an increase in accuracy" for music classification tasks and gives an intuition of what the network may be learning; showing that musically motivated architectures may be beneficial for MIR. Moreover, Choi *et al.* [1] proposed a method called auralisation which is an extension of the CNNs visualization method [22]. Thus, it allows to interpret by listening what each CNN filter has learned.

[1] http://benanne.github.io/2014/08/05/spotify-cnns.html

978-1-4673-8695-1/16/$31.00 ©2016 IEEE

Despite the efforts on trying to puzzle out what the networks are learning, it is still not clear how to navigate through the network parameters space. It is hard to discover the adequate combination of parameters for a particular musical task, which leads to architectures being difficult to interpret. Given this, our work aims to rationalize this process by proposing musically motivated architectures. Specially, we study how CNNs can be tailored towards learning generalizable musical concepts. For doing so, in section II we conceptually discuss what CNN filters with different shapes can learn and, in section III and IV, we validate such concepts on experiments. Several architectures are evaluated against a dataset that is known for having classes that are already well represented by (solely) its tempo: the Ballroom dataset. The special characteristics of this dataset, allow us assessing musically inspired CNNs architectures. Section V concludes and points out future work.

II. Motivations

A. Audio material

Experiments are realized using the Ballroom dataset[2]: 698 tracks, around 30 seconds long, divided into 8 music genres: cha-cha-cha, jive, quickstep, rumba, samba, tango, viennese-waltz and slow-waltz. Despite its known shortcomings [6] [20], this dataset has been used extensively and it allows to:

- evaluate our algorithm comparing it with state-of-the-art results: Marchand *et al.* [11] achieved 93.12% accuracy predicting the Ballroom classes – without using BPM annotations. We set this algorithm as the baseline for our deep learning methods using time-frequency features because Marchand *et al.* take advantage of time and frequency cues, as well.
- understand the musical characteristics of the Ballroom dataset. Even though this dataset was originally designed to study rhythmic patterns, Guyon *et al.* [6] showed that each Ballroom class is already rather well characterized by its tempo. A k-nearest neighbor (with k=1) using the BPM annotations achieved 82.3% accuracy. Therefore, tempo and rhythm are relevant when predicting the Ballroom classes.

We expect our network to learn such relevant temporal dependencies from data. In particular, we propose a CNN architecture (*Time*) specifically designed to fit those. For fair comparison, we consider Guyon *et al.* [6] as a baseline for these methods only using temporal features. We also propose another architecture (*Frequency*) that is designed not to learn such relevant temporal dependencies. For these methods, we set a random baseline based on the probability of guessing the most likely Ballroom class, cha-cha-cha: 15.9 %. Therefore, the Ballroom dataset allows us assessing musically motivated architectures; for that reason no more datasets are used.

The audio is fed to the network through fixed-length mel-spectrogram samples [15], *N* frames wide. Interestingly, Dieleman *et al.* [4] input raw audio to the network and found that the lowest convolutional layer was learning "frequency-selective features covering the lower half of the spectrum", similarly to what a mel filter-bank does, what motivates our choice.

Throughout this work we use 40 bands mel-spectrograms derived from a STFT-spectrogram computed with a Blackman Harris window of 2048 samples (50% overlap) at 44.1 kHz. Phases are discarded.

B. Deep learning

Several architectures can be combined to construct deep learning algorithms: feed-forward neural networks, RNNs or CNNs. However, since the goal of our work is to tailor the network towards learning musically relevant features, CNNs seemed an intuitive choice regarding that the input data is formatted as a spectrogram. CNNs fed with spectrograms allow the design of CNNs filters having interpretable dimensions in the first layer: time and frequency, this allows designing musically motivated architectures.

1) Filter shapes: Due to the CNNs success in the image processing research field, its literature significantly influenced the MIR community. In the image processing literature, squared small CNNs filters (*i.e.* 5x5 or 12x12) are common [22]. As a result of that, MIR researchers tend to use similar filter shape setups [1] [17] [14]. However, note that the image processing filter dimensions have spatial meaning, while the audio spectrograms filters dimensions correspond to time and frequency. Therefore, *wider* filters may be capable of learning longer temporal dependencies in the audio while *higher* filters may be capable of learning more spread timbral features. In order to motivate researchers to be conscious about the potential impact of choosing one filter shape or another, three examples and a use case are discussed in the following. Throughout this manuscript we assume the spectrogram dimensions to be M-*by*-N and the filter dimensions to be m-*by*-n. M and m standing for the number of frequency bins and N and n for the number of time frames:

- Squared/rectangular filters (*m-by-n* filters) are capable of learning time and frequency features at the same time. This kind of filter is one of the most used in the MIR literature. Such filters can learn different musical aspects depending on how m and n are set. For example, the bass could be well modeled with a small filter ($m \ll M$ and $n \ll N$, representing a sub-band for a short-time) because: this instrument is sufficiently characterized by the lower bands of the spectrum and the temporal evolution of the bass notes is not so long. An interesting interpretation of such small filters is that they can be considered pitch invariant to some extent. Since the convolution also happens in the frequency domain, such filters can be modulated to represent different pitches. However, note that such pitch invariability would not hold for instruments having a large pitch range since the timbre of an instrument changes accordingly to its pitch. As another example, cymbals or snare drums – that are broader in frequency with a fixed decay time– could be suitably modeled setting $m = M$ and $n \ll N$.

[2]http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html

Please note that a bass could also be modeled with this filter but the pitch invariance interpretation will not hold because its dimensions ($m = M$) do not allow the filter to convolve along frequency and therefore, the pitch will be encoded together with the timbre.

- Temporal filters (*1-by-n*): setting the frequency dimension m to 1, such filters will not be capable of learning frequency features but will be specialized on finding temporal dependencies relevant for the task to be learned from the training data. However, note that even though the filters themselves are not learning frequency features, upper layers may be capable of exploiting frequency relations – the frequency interpretation still holds for the resulting activations because the convolution operation is done bin-wise (m=1). From the musical perspective, one expects those temporal filters to learn relevant rhythmic/tempo patterns within the analyzed bin.
- Frequency filters (*m-by-1*): setting the time dimension n to 1, such filters will not be capable of learning temporal features but will be specialized on modeling frequency features relevant for the task to be learned from the training data. Similarly as for the temporal filters, upper layers could still find some temporal dependencies from the resulting activations. From the musical perspective, one expects these frequency filters to learn pitch, timbre or equalization setups, for example.

To conclude this section, we would like to discuss the results of Choi *et al.* [1][3] as a use case. They use a 4-layer CNN of squared *12-by-12* filters. After auralising and visualizing the network filters, they conclude that their deep learning algorithm was capable of: finding attack/onsets, selecting bass notes and separating kick drums. As previously discussed, squared small filters may be capable of modeling instruments appearing in a sub-band (bass and kick) and also to model temporal features (onsets) due to its length. However, what would be a surprise is to observe that such a network is modeling cymbals or snare drums, what may be definitely challenging for a CNN with such filter shapes.

III. Experiments

Three architectures are introduced to experiment with filter shapes that are designed to fit several music concepts:

1) *Black-box* architecture. This system is based on previous work using a *m-by-n* CNN filter architecture for the task of music classification [10]. In this setup –obtaining the best results for the MIREX 2015 task of music/speech classification[4]– a single convolutional layer with 15 *12-by-8* filters and a *2-by-1* max-pool layer has been used, followed by a feed-forward layer of 200 units connected to the output softmax layer. We adapted this approach to further get better accuracy results changing slightly the setup to 32 *12-by-8* filters and a max-pool layer of *4-by-1* (*cf.* Figure 1). Such tests are not discussed throughout this manuscript because this is not the focus of the paper.
2) *Time* architecture is particularly designed to learn temporal dependencies. It is composed by a convolutional layer of 32 temporal filters (*1-by-n*) followed by a max-pool layer of (*M-by-1*) connected to the output layer (*cf.* Figure 2). The fact that the max-pool layer spans all over the frequency axis (*m=M*) and covers only one frame (*n=1*), allows: propagating only temporal content due to the summarization done among frequencies and preserving the frame resolution, respectively.
3) *Frequency* architecture is designed to learn frequency features. It is composed by a convolutional layer of 32 frequency filters (*m-by-1*) followed by a max-pool layer of (*1-by-N*) connected to the output layer (*cf.* Figure 3). The max-pool layer (with *n=N*) operates similarly as in the *Time* architecture, but in that case the summarization is in time. Note that the extreme case of a *Frequency* architecture would be to input only one frame to the network; however, we expect the statistics provided by the max-pool layer to help the network learning timbral cues.

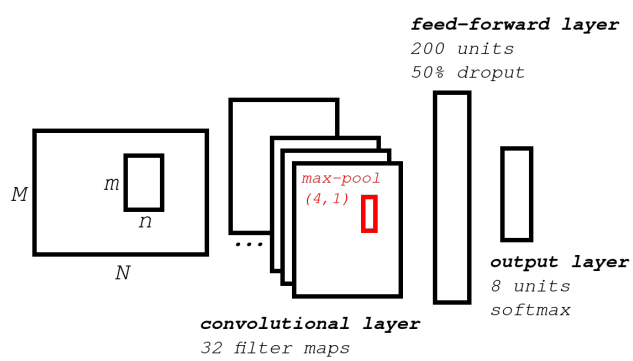

Fig. 1. Schema of the *Black-box* architecture.

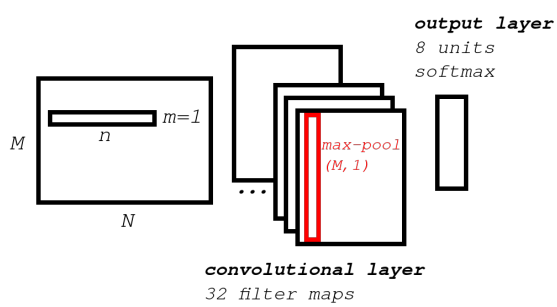

Fig. 2. Schema of the *Time* architecture.

Black-box follows the standard architecture that one can find in the literature [1] [14]. We call it black-box because there is no musically motivated reason for such architectural choices. Note that the feature maps resulting of the convolutional layer are difficult to interpret because there is not apparent motiva-

[3]http://keunwoochoi.blogspot.com.es/2015/10/ismir-2015-lbd.html
[4]http://www.music-ir.org/mirex/wiki/2015:Music/ /Speech_Classification_and_Detection_Results

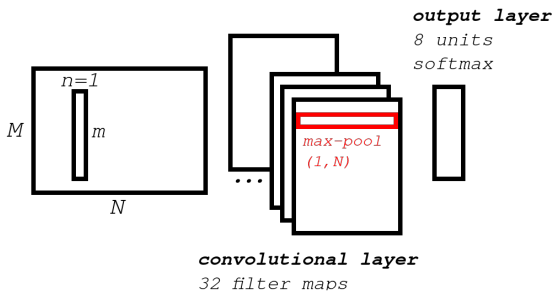

Fig. 3. Schema of the *Frequency* architecture.

tion behind. *Time* and *Frequency* are introduced as musically inspired systems. No additional feed-forward layer (as in the *Black-box*) is put before the output layer to control that all the resulting feature maps are interpretable. Considering that the networks are trained using the Ballroom dataset, the *Time* architecture may be capable of learning tempo/rhythm features and the *Frequency* architecture may be capable of learning relevant timbral cues.

Experimenting with the filter shapes of the previously presented architectures, we discuss several musically motivated choices:

1) *Filter length for learning temporal dependencies:* One of the goals of our work is to model the relevant temporal dependencies within the Ballroom dataset. In this subsection we focus on tempo. Short temporal filters may have difficulties on learning slow tempos, two filter lengths are studied for the *Time* architecture:
   - 60 frames ($\approx$ 1.4 sec). Only 1 beat can be accommodated by the filter for the slowest tempo in the Ballroom dataset, 60 BPMs. However, the filter can learn about 5 beats for the fastest tempo in the Ballroom dataset, 224 BPMs.
   - 200 frames ($\approx$ 4.6 sec). Up to 4 beats can be learned by such filter for the slowest tempo, 60 BPMs. However, the filter can fit 17 beats for the fastest tempo in the Ballroom dataset, 224 BPMs.

   For the *Black-box* architecture we study two filter lengths: $n = 8$ and $n = 200$. The former is based on previous work where the filter shape was optimized for estimating the Ballroom classes while the latter is motivated by the previous discussion. For experiments with $n = 8$ and $n = 60$, the input spectrogram is set to 80 frames. However, for experiments with 200 frames filters, the input is enlarged to 250 frames. As result of increasing the number of frames available for each spectrogram, less spectrogram segments are sampled per track, meaning that less training examples are available although we are using the same dataset.
2) *Pitch invariant filters:* Even though the pitch invariability of such filters still needs to be proven formally, intuitively it seems beneficial that the filter can convolve in frequency ($m < M$). In that way, it can learn frequency features that are less pitch dependent and therefore, the frequency filters are capable of learning more general concepts (*i.e.* timbre). Within the *Frequency* architecture experiments, we asses several frequency filter shapes (setting m differently) to study how pitch invariant filters behave when predicting the Ballroom classes.

Finally, we join in the same model the *Time* and *Frequency* architectures that achieved better accuracies predicting the Ballroom classes in previous experiments. Figure 4 depicts an schema of this combined architecture. The motivation behind this model is to join the *Time* and *Frequency* architectures, that are learning complementary aspects from the data, to create a more expressive musically motivated architecture. A feed-forward layer of 200 units is set on top of the *Time* and *Frequency* architectures to allow the network making use of time and frequency features jointly. We experiment with two setups, learning from: (1) random initialization or (2) weights initialized with the previous *Time* and *Frequency* most successful models. We refere to these architectures as (1) *Time-Frequency* and (2) *Time-FrequencyInit*, respectively. The *Time* and *Frequency* parts in the *Time-FrequencyInit* combined network are initialized using the best model for *Time* and *Frequency* architectures, respectively. Each initialization considers its corresponding model trained earlier in the same fold to avoid training/testing with the same data.

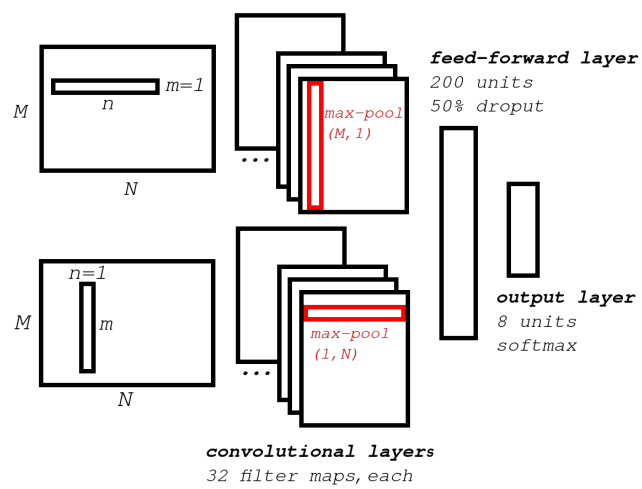

Fig. 4. Schema of the *Time-Frequency* & *Time-FrequencyInit* architectures.

A. Experimental setup

A dynamic range compression is applied to the input spectrograms element-wise in the form of $log(1 + C \cdot x)$ where $C = 10.000$ is a constant controlling the amount of compression [4]. The resulting spectrograms are normalized so that the whole dataset spectrograms (together) have zero mean and variance one. Note that this normalization is not attribute-wise, as this would perturb the relative information encoded between spectrogram bins/frames. The activation functions of the hidden layers are linear rectifiers [12] (ReLUs) with a final 8-way softmax, where each output unit corresponds to

a Ballroom class. 50% dropout [19] is applied to the feed-forward layers. The output unit having the highest output activation is selected to be the model's class prediction. Each network is trained using minibatch gradient descent with minibatches of 10 samples, minimizing the categorical cross-entropy between predictions and targets for each sample. It is trained from random initialization using an initial learning rate of 0.01, unless said explicitly. The learning rate is divided by two every time the training loss gets stuck, *i.e.* when it does not improve for 40 epochs. The model reporting better accuracy in the validation set is kept as the best model (a variant of early-stopping [16]) to be evaluated in the test set for accuracy report. All experiments are developed using Lasagne –a Theano-based framework allowing GPU acceleration– and are available online[5].

B. Evaluation

Accuracies are computed using 10-fold cross validation with a randomly generated train-validation-test split of 80%-10%-10%. Since the input spectrograms are shorter than the total length of the song spectrogram, several estimations for each song can be done. A simple majority vote approach serves to decide the estimated class.

IV. Results & Discussion

Results are presented in Table I.

The *Black-box* architecture reached inferior accuracy results than the state-of-the-art provided by Marchand *et al.* [11].

The *Time* architecture is capable of achieving similar results as Guyon *et al*. This result provides evidence that it is important to first understand the training datasets used by our deep learning algorithms. Doing so, researchers should be able to use such knowledge to design architectures that better fit the problem. This is specially relevant for the MIR field since it has already been pointed out that machine learning algorithms are learning how to "reproduce the ground truth" rather than learning musical concepts [21]. Tackling deep learning architectures in such a musical way, may reduce that risk and will increase the capability of the systems to learn musically relevant features. As a final reflection, note that even it is clear that the temporal filters are learning relevant temporal dependencies, we can not claim that these are tempo or rhythm. Further research is needed to assess that with experiments that tackle directly this issue.

Interestingly, the *Frequency* architecture is capable of learning discriminative frequency features from the data. It clearly outperforms its baseline, denoting that the frequency features are more relevant for predicting the Ballroom classes than expected. However, this architecture does not outperform the others since it was not designed to learn temporal dependencies, that are capital for the Ballroom classes differentiation.

Wider filters (*n=200*) do not estimate better the Ballroom classes than *shorter* ones (*n=60*). This result is surprising because it seems a challenging task, even for a human, to discriminate the tempo with such short sounds – less than 2 seconds. Two plausible reasons exist to explain that *shorter* filters are estimating better the Ballroom classes: (1) predicting the Ballroom classes do not mean explicitly predicting tempo and (2) less training examples are available because for using longer filters we need to input longer spectrograms to the network, which decreases the number of sampled spectrograms per track. Exploring data augmentation paradigms may be interesting to improve accuracy results for longer filters. In fact, data augmentation is a powerful tool where musically motivated choices could be done. For example, Nam *et al.* [13] proposed a method called *onset-based sampling*, that samples tracks considering the onset times instead of sampling arbitrarily. Thus, allows sampling the data in a musically meaningful way that allows overlapping. Such *overlapping onset sampling* may be an interesting data augmentation strategy to be considered. However, many other musically inspired data augmentation strategies could be adopted: pitch shifting, time stretching [18] or even re-mixing [7].

Designing the filters such that they can convolve in frequency ($m < M$), helps predicting the Ballroom classes. This probably prevents the filters to learn individual pitches centering its capacity on modeling timbre, what allows the network to be more expressive. Also note that the performance improves when reducing m down to 32. We speculate that small filters ($m < 32$) may have difficulties on learning timbral features. Finally, note that this *pitch invariance* also happens in a similar way for the *Time* architecture. Allowing the filter to convolve along time, the network is able to fit the filter at the point in time where the beats are happening. Thus, it could be considered as a time-position invariant filter as well.

The last block of experiments show that the musically motivated *Time-Frequency* architectures can achieve similar results as *Black-box* approaches. However, *Time-Frequency* architectures are more interpretable since they were designed for having under control what the network is passing through layers. We expect these musically motivated architectures to be more treatable, as they should allow researchers to dig into what the networks have learned in a more intuitive way. Moreover, the *Time-FrequencyInit* architecture estimates slightly better the Ballroom classes than *Time-Frequency*, denoting that pre-initializing the networks is beneficial. Thus, it allows to start the optimization problem closer to a minimum with stronger generalization properties.

Finally, we also want to remark the importance of the training datasets for deep learning experiments. The here presented work was possible because the musical characteristics of the Ballroom dataset are well known. Thus, it allowed us to address the architectures and experiments design in a musical way. Datasets with musicological description and with finer annotations are indeed necessary for the advance of the MIR state-of-the-art.

V. Conclusions and future work

We have shown how CNNs can be designed having musical aspects in mind. The network was trained on spectrograms,

[5]http://github.com/jordipons/CBMI2016

TABLE I

FOUR ARCHITECTURES ARE STUDIED: *Black-box*, *Time*, *Frequency* AND *Time-Frequency*. EACH ONE IS PRESENTED IN A BLOCK, DIVIDED BY HORITZONTAL LINES. *Black-box* AND *Time* STUDY DIFFERENT FILTER LENGTHS. *Frequency* STUDY PITCH INVARIANT FILTERS. *Time-Frequency* STUDY DIFFERENT INITIALIZATION SCHEMES.

Architecture	Input (M,N)	Filter shape (m,n)	# param.	Max-pool	Accuracy: mean ± std 10 cross-fold validation	Baseline
Black-box	(40,80)	(12,8)	3.275.312	(4,1)	**87.25** ± 3.39 %	93.12 % → Marchand *et al.* [11]
Black-box	(40,250)	(12,200)	2.363.440	(4,1)	82.80 ± 5.12 %	93.12 % → Marchand *et al.* [11]
Time	(40,80)	(1,60)	7.336	(40,1)	**81.79** ± 4.72 %	82.3% → Guyon *et al.* [6]
Time	(40,250)	(1,200)	19.496	(40,1)	81.52 ± 3.87 %	82.3% → Guyon *et al.* [6]
Frequency	(40,80)	(30,1)	3.816	(1,80)	59.45 ± 5.02	15.9 % → Most probable class
Frequency	(40,80)	(32,1)	3.368	(1,80)	**59.59** ± 5.82	15.9 % → Most probable class
Frequency	(40,80)	(34,1)	2.920	(1,80)	58.17 ± 3.58	15.9 % → Most probable class
Frequency	(40,80)	(36,1)	2.472	(1,80)	57.88 ± 5.38	15.9 % → Most probable class
Frequency	(40,80)	(38,1)	2.024	(1,80)	57.45 ± 5.93	15.9 % → Most probable class
Frequency	(40,80)	(40,1)	1.576	(1,80)	52.43 ± 5.63	15.9 % → Most probable class
Time-Frequency	(40,80)	(1,60)-(32,1)	196.816	(40,1)-(1,80)	86.54 ± 4.29	93.12 % → Marchand *et al.* [11]
Time-FrequencyInit	(40,80)	(1,60)-(32,1)	196.816	(40,1)-(1,80)	**87.68** ± 4.44	93.12 % → Marchand *et al.* [11]

thus the CNN filter dimensions are interpretable in time and frequency. We have used such observation to discuss how several filter shapes can model musical aspects. Furthermore, we have proposed some musically motivated deep learning architectures and we have shown that these can achieve competitive results on predicting the Ballroom dataset classes. However, this work is only a step towards understanding what the deep neural networks are modeling. As future work it is planned to analyze what these musically motivated architectures have learned, specially we would like to do experiments studying the pitch invariant filters and to observe whether the trained filters have learned tempo/rhythm or not.

ACKNOWLEDGMENTS

The Titan X used for this research was donated by the NVIDIA Corporation. This work is partly supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502). We would like to thank Marius Miron for many helpful discussions.

REFERENCES

[1] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim. Auralisation of deep convolutional neural networks: Listening to learned features. In *16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015.

[2] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 669–674. University of Miami, 2011.

[3] Sander Dieleman and Benjamin Schrauwen. Multiscale approaches to music audio feature learning. In *14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 116–121. Pontifícia Universidade Católica do Paraná, 2013.

[4] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, 2014.

[5] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[6] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *AES 25th International Conference*, pages 196–204, 2004.

[7] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2136–2147, 2015.

[8] Eric J Humphrey, Juan P Bello, and Yann LeCun. Feature learning and deep architectures: new directions for music informatics. *Journal of Intelligent Information Systems*, 41(3):461–481, 2013.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[10] Thomas Lidy. Spectral Convolutional Neural Network for Music Classification. In *Music Information Retrieval Evaluation eXchange (MIREX)*, Malaga, Spain, 2015.

[11] Ugo Marchand and Geoffroy Peeters. The modulation scale spectrum and its application to rhythm-content description. In *DAFx*, pages 167–172, 2014.

[12] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[13] Juhan Nam, Jorge Herrera, and Kyogu Lee. A deep bag-of-features model for music auto-tagging. *arXiv preprint arXiv:1508.04999*, 2015.

[14] Taejin Park and Taejin Lee. Music-noise segmentation in spectrotemporal domain using convolutional neural networks. In *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[15] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification), CUIDADO technical report. 2004.

[16] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998.

[17] Jan Schluter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983, 2014.

[18] Jan Schlüter and Thomas Grill. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In *16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[20] B.L. Sturm, C. Kereliuk, and A. Pikrakis. A closer look at deep learning neural networks with low-level spectral periodicity features. In *4th International Workshop on Cognitive Information Processing (CIP)*, pages 1–6, May 2014.

[21] Bob L Sturm. A simple method to determine if a music information retrieval system is a horse. *IEEE Transactions on Multimedia*, 16(6):1636–1644, 2014.

[22] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV*, pages 818–833. Springer, 2014.

[23] Xinquan Zhou and Alexander Lerch. Chord detection using deep learning. In *16th International Society for Music Information Retrieval Conference (ISMIR)*, volume 53, 2015.