TimeScaleNet : a Multiresolution Approach for Raw Audio Recognition using Learnable Biquadratic IIR Filters and Residual Networks of Depthwise-Separable One-Dimensional Atrous Convolutions

Eric Bavu, Aro Ramamonjy, Hadrien Pujol, Alexandre Garcia

TimeScaleNet : a Multiresolution Approach for Raw Audio Recognition using Learnable Biquadratic IIR Filters and Residual Networks of Depthwise-Separable One-Dimensional Atrous Convolutions

Éric Bavu*, *IEEE SPS Member*, Aro Ramamonjy, Hadrien Pujol, and Alexandre Garcia,

Special Issue on Data Science: Machine Learning for Audio Signal Processing

***Abstract*—In the present paper, we show the benefit of a multi-resolution approach that allows to encode the relevant information contained in unprocessed time domain acoustic signals. TimeScaleNet aims at learning an efficient representation of a sound, by learning time dependencies both at the sample level and at the frame level. The proposed approach allows to improve the interpretability of the learning scheme, by unifying advanced deep learning and signal processing techniques. In particular, TimeScaleNet's architecture introduces a new form of recurrent neural layer, which is directly inspired from digital IIR signal processing. This layer acts as a learnable passband biquadratic digital IIR filterbank. The learnable filterbank allows to build a time-frequency-like feature map that self-adapts to the specific recognition task and dataset, with a large receptive field and very few learnable parameters. The obtained frame-level feature map is then processed using a residual network of depthwise separable atrous convolutions. This second scale of analysis aims at efficiently encoding relationships between the time fluctuations at the frame timescale, in different learnt pooled frequency bands, in the range of [20 ms ; 200 ms]. TimeScaleNet is tested both using the Speech Commands Dataset and the ESC-10 Dataset. We report a very high mean accuracy of $94.87 \pm 0.24\%$ (macro averaged F1-score : $94.9 \pm 0.24\%$) for speech recognition, and a rather moderate accuracy of $69.71 \pm 1.91\%$ (macro averaged F1-score : $70.14 \pm 1.57\%$) for the environmental sound classification task.**

***Index Terms*—Machine hearing, Audio recognition, Learnable Biquadratic filters, Deep Learning, Time domain modelling, Multiresolution**

I. Introduction

In early years of machine hearing, conventional recognition tasks involved hand-crafted features [1], [2] such as Mel-frequency cepstral coefficients (MFCCs) [3] or Perceptual Linear Prediction coefficients (PLPs) [4] as inputs to the developed models. The rise of deep learning algorithms based on convolutional neural network – along with their ability to learn from localized patterns in two-dimensional maps – led to the use of time-frequency representations based on short-time Fourier transforms as the most common choice of input for machine hearing tasks. However, there is still no consensus on the best representation to use in order to better encode the information needed to recognize sounds, since the parameters heavily depends on the type of sound to be classified, and differ greatly for sound event detection, speech recognition, music classification or environmental sound recognition [5]–[10].

Since the unprocessed, time-domain audio signals contain all the information to be extracted for the machine hearing task, the scientific community has recently put some efforts to directly use the raw waveforms as inputs for deep learning models [11]–[16]. Acoustic model learning from the raw waveform has therefore emerged as an active area of research in the last few years, and recent works have shown that this approach allows to successfully learn the temporal dynamics scales of the waveforms. While they show promising results, the models mostly use large filters, which can model passband filters [14] approximating time-domain cochlear filter estimates.

These studies, along with recent advances in machine learning architectures for one-dimensional signals [17]–[19] has motivated the present work, which aims at showing the benefit of an efficient multi-resolution approach for machine hearing, that allows to avoid the need to pre-process the waveforms in order to encode the relevant information contained in the acoustic signal. The proposed approach avoids using large convolutional kernels, by introducing a new form of recurrent neural cell, directly inspired from IIR digital signal processing. The proposed deep neural network aims at learning an efficient representation of a sound, by specializing at both the sample level and the frame level. In the following, TimeScaleNet's architecture is detailed, and its links with digital signal processing and cognitive models are highlighted. Its performances for sound recognition are detailed for both speech recognition on a keyword spotting task, and environmental sound recognition. We also derive and analyze the learnt equivalent

É. Bavu, A. Ramamonjy, H. Pujol and A. Garcia are with the Laboratoire de Mécanique des Structures et des Systèmes Couplés (LMSSC), Conservatoire national des arts et métiers (Cnam), 292 rue Saint-Martin, 75003 Paris, France.

* Corresponding author e-mail: eric.bavu@lecnam.net

Manuscript received October 01, 2018. - First Revision sent January 31, 2019 - Second Revision sent March 23, 2019 - Accepted for publication March 23, 2019

filterbank magnitudes in order to give further interpretability of the machine hearing process in the scope of auditory filters models.

II. METHODS

The proposed method takes a raw audio waveform as input for a multi-class classification task. The global neural network architecture is detailed in II-A. As shown on Fig. 1, this architecure can be split in two major subnets, aiming at extracting relevant features from the raw waveform at two different timescales. The architecture and the detailed implementation of these two subnets are explained in II-B and II-C. The training procedure is also detailed in II-D.

A. *Global neural network architecture*

In the present section, we detail the neural network model we use for our experiments. In the following, the global neural network will be referred as TimeScaleNet, in reference to the fact that our model aims at optimizing the learnt representation of raw audio waveforms, at two different timescale levels.

As shown on Fig. 1, the first subnet of TimeScaleNet's architecture is called BiquadNet (see II-B), in reference to the similarity between its first layer and the standard biquadratic filters in digital signal processing. BiquadNet acts at the sample level, and aims at encoding the information for time scales in the range of [100 μs ; 20 ms], corresponding to a frequency range of [50 Hz ; 10 kHz]. This learnable IIR filterbank allows to compute a time-frequency-like representation, that is fed to the next subnet of our architecture. The first layer of BiquadNet is a non-conventional recurrent neural network (RNN) layer, in comparison to vanilla RNNs [20], standard Gated Recurrent Units (GRU) [21], or Long Short Term Memory (LSTM) layers [22], whose architectures have less similarities with standard digital signal processing than the proposed layer. The proposed "biquadratic" RNN filter can be thought as a set of infinite impulse-response (IIR) filters, expressed as a biquadratic filterbank [23]. Digital biquadratic filterbanks have already been used in the signal processing literature for the modelling of the human auditory function [24], [25]. However, to the best of author's knowledge, this is the first time that a Deep neural network uses a biquadratic-form RNN layer with learnable coefficients, that self-adapts to the audio dataset that has to be classified. The proposed approach allows a computationally-efficient IIR bandpass filtering, using only two learnable parameters for arbitrarily long receptive fields, rather than 1-dimensional convolutional neural networks with wide kernels. In previous studies, authors reported the use of large one-dimensional convolutions as equivalent of FIR bandpass filtering, in order to approximate perceptual filterbanks – such as a gammatone filterbank [14], [16], [26]. The overall output of BiquadNet is a two dimensional map, where the first dimension represents different pooled frequency channels, since the last layer of BiquadNet is a pointwise convolution which aims at aggregating different frequency bins together in order to better encode vowels formants and consonants. The second dimension represents overlapping frames, where an energy-like feature is computed

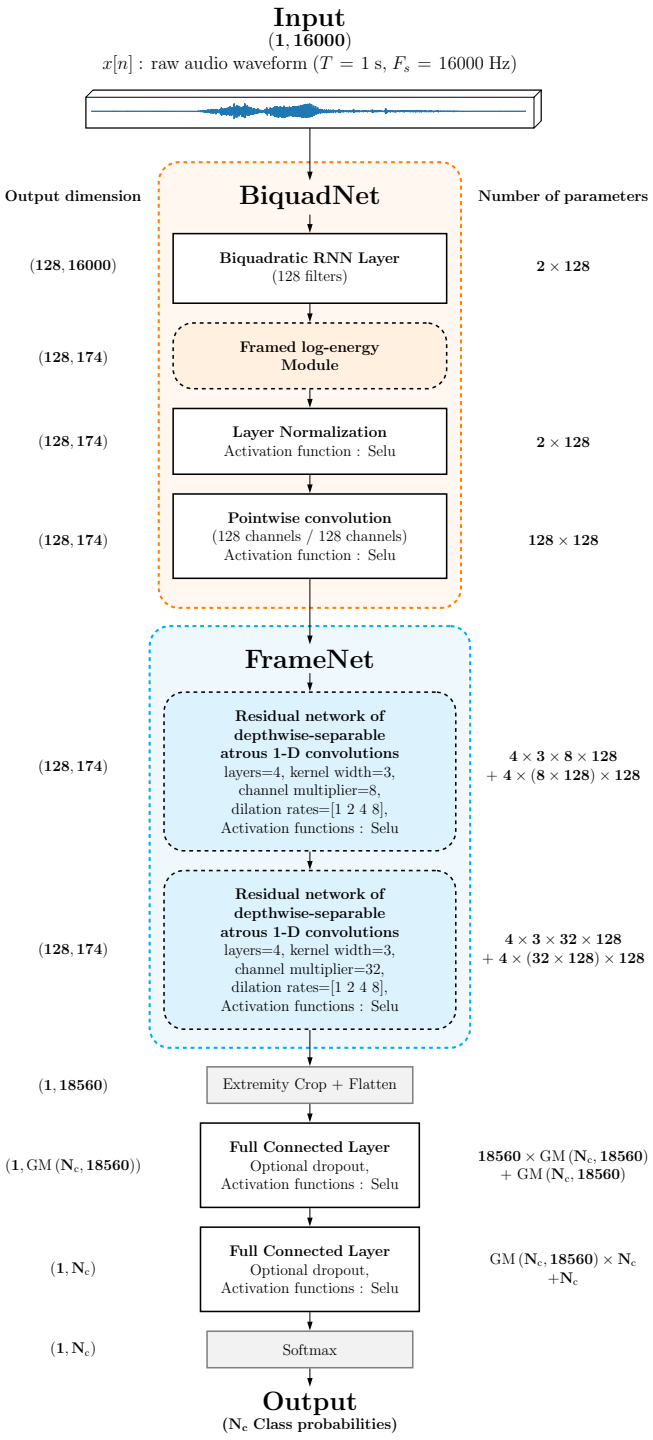


Fig. 1. (Color online) Schematical representation of the global architecture of TimeScaleNet. This neural network takes a raw waveform as input. The overall architecture aims at optimizing the learnt representation at two timescales levels (see II-B and II-C for more details on BiquadNet and FrameNet). On the left (resp. on the right) of each subnets, the output dimensions (resp. the number of learnable parameters, depending on the number of classes) are given for each subnet. $GM(N_1, N_2)$ stands for geometric mean of N_1 and N_2. For a 10-class recognition task, the total number of learnable parameters is 10.7×10^6

by the subnet. The overall architecture of BiquadNet and its implementation are detailed in II-B.

The obtained time-frequency-like representation at the output of BiquadNet is then fed to the second subnet, referred in the following as "FrameNet" (see II-C), because it acts at the frame level, in order to efficiently encode the time fluctations in the range of [20 ms ; 200 ms]. This second scale of analysis aims at extracting the relevant relationships between time fluctuations in different learnt pooled frequency channels, with a large receptive field. For this purpose, we propose the use of residual networks of one-dimensional depthwise separable atrous convolutions, which allow to operate on channel-wise frames in a computationally efficient way.

FrameNet shares some of the characteristics of the SliceNet architecture, recently introduced by Kaiser et al. [18] for neural machine translation. The main ingredients of FrameNet are stacked residual atrous convolutions, which have already been recently emerged as an efficient architecture for audio generation [17] and denoising [19]. Each depthwise separable convolutional layer is followed by a Selu nonlinear activation [27], which has been introduced in the literature in order to avoid standard batch normalization processes, without degrading the computational efficiency of deep neural networks. In comparison to RELU, the Selu activation has self-normalizing properties, because the activations that are close to zero mean and unit variance, propagated through many network layers, will converge towards zero mean and unit variance. This, in particular, makes the learning highly robust and allows to train networks that have many layers. We also use residual connections between each depthwise separable convolutional layers, in order to allow the network to be deeper without impacting accuracy and vanishing gradients problems [28]. The overall architecture of FrameNet and its implementation are detailed in II-C.

The use of residual connections between each atrous depthwise separable convolutional layer requires that the output of each layer has the same dimension as the overall output of BiquadNet. As a consequence, each atrous convolution is computed using zero-padding. At the end of FrameNet however, in order to keep the overall portion of the output which is valid, *i.e.* not using any padding zeros, the output of FrameNet is then cropped in the timeframe dimensions, therefore only keeping the time frames corresponding to the to the valid part for all the atrous convolutional layers used in FrameNet. The obtained map is then flattened, and fed to two full-connected layers with Selu activations and optional dropout, in order to compute a vector of dimension N_c representing the probability of belonging to the classes of the dataset.

B. BiquadNet architecture : raw waveform processing

As introduced in the previous subsection, from machine-learning point of view, the first layer of BiquadNet is a non-conventional recurrent neural network cell. From a digital signal processing point of view however, this RNN cell is directly derived from a widely used infinite impulse response (IIR) filter architecture. In digital signal processing, IIR filters are the most efficient type of filter to implement, because they require less computation and memory than FIR filters in order to perform similar filtering operations. However, IIR filters present the main disadvantage of having a nonlinear phase response. We address this problem by implementing a bidirectional biquadratic RNN cell, which allows to achieve forward-backward filtering [29], [30], in order to perform a perfect zero-phase filtering in the time domain. The other main disadvantage of IIR filters is their potential numerical instability : high-order IIR filters can be highly sensitive to quantization of their coefficients, and can easily become unstable. The use of first and second-order IIR filters only makes the stability problem more tractable. This is the main reason why most digital signal processors implement stacks of biquadratic IIR filters. This kind of topology can be easily transposed to machine learning, where deep neural network topologies often use stacking of similar layers. In the following, we will use the normalized direct-form I of biquadratic filters, which have the following difference equation (1), which defines the value of the current output value $y[n]$ at sample n, using the current input value $x[n]$ and the two previous values of the output and the input :

$$\begin{aligned} y[n] = b^{(0)}x[n] + b^{(1)}x[n-1] + b^{(2)}x[n-2] \\ -a^{(1)}y[n-1] - a^{(2)}y[n-2] \end{aligned} \tag{1}$$

Using the Z-transform, this filter exhibits two zeros and two poles, and corresponds to the ratio of two biquadratic functions, as shown in equation (2):

$$H(z) = \frac{b^{(0)} + b^{(1)}z^{-1} + b^{(2)}z^{-2}}{1 + a^{(1)}z^{-1} + a^{(2)}z^{-2}} \tag{2}$$

This learnable biquadratic filter structure has been implemented using the Tensorflow open source software library [31]. The chosen implementation corresponds to a Direct-Form I [30], which can be represented as the flow graph depicted on Fig. 2. This flow graph also explicitly shows the adjustable parameters $(b_i^{(0)}, b_i^{(1)}, b_i^{(2)}, a_i^{(1)}, a_i^{(2)})$ used in each RNN cells of BiquadNet.

Using (2), the stability of biquadratic filters is ensured if and only if $a^{(1)}$ and $a^{(2)}$ are inside the "stability triangle" [32] depicted on Fig. 3. Since we aim at obtaining a "time-frequency"-like representation at the output of BiquadNet, we restrict the possible values of the coefficients of the learnable IIR filterbank to correspond to passband versions of a biquadratic IIR filter. This allows to simplify the stability properties of the learnt filters, since passband biquadratic

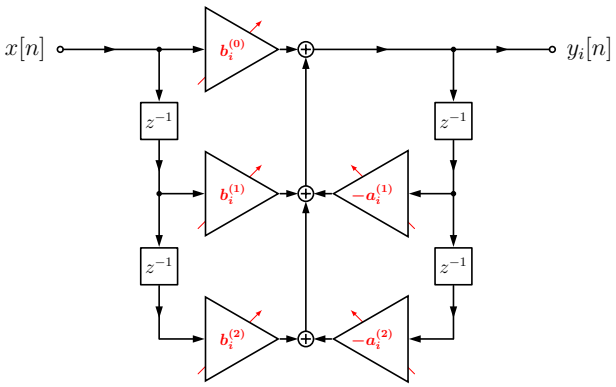


Fig. 2. (Color online) Flow graph of the learnable biquadratic infinite impulse response filters used in the proposed BiquadNet. $x[n]$ is the time domain waveform input, $y_i[n]$ is the i[th] output of the filterbank. The slanted arrows behind gains $(b_i^{(0)}, b_i^{(1)}, b_i^{(2)}, a_i^{(1)}, a_i^{(2)})$ indicate that these parameters are adjustable (learnable).

filters are unconditionally stable. However, for floating point implementations, the quality factor of digital passband filters is usually restricted in order to avoid numerical instabilities when approaching the boundaries of the stability triangle. It is also particularly interesting to note that passband biquadratic filters (also referred as two-poles two-zeros filters in the literature) have been demonstrated to be good numerical models of auditory filterbanks [24], [25], where the quality factors of perceptual filters match a viable stability region, even for floating point implementations.

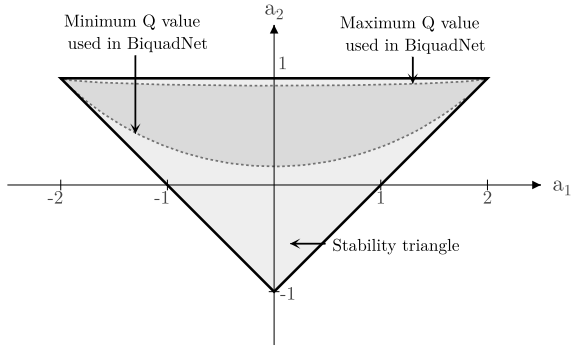


Fig. 3. Stability triangle of a biquadratic filter. In order to be stable, the coefficients $a^{(1)}$ and $a^{(2)}$ values should respect a set of inequalities that correspond to the depicted light-grey zone. In BiquadNet, we implement learnable passband biquadratic filters, with constraints on both the central frequency f_c and the quality factor Q. The corresponding learnt values of $a^{(1)}$ and $a^{(2)}$ are in the depicted dark grey zone, therefore ensuring that the learnt IIR filters are numerically stable, even with floating point precision.

Each biquadratic bandpass filter of the learnable filterbank represented by the biquadratic RNN layer can be fully determined using only two parameters, $K^{(i)} = \tan\left(\pi f_c^{(i)}/f_s\right)$ and $Q^{(i)}$, where f_s is the sample frequency, $f_c^{(i)}$ is the central frequency of the i[th] bandpass filter, and $Q^{(i)}$ is the quality factor of the i[th] bandpass filter. $f_c^{(i)}$ and $Q^{(i)}$ physically represent the exact same quantities as in analog, second-order bandpass filters, and can be linked to models of auditory filterbanks [24], [25]. The parameter $K^{(i)}$ is derived from the bilinear transformation with frequency warping compensation [30] in order to compute the coefficients of the equivalent digital second order bandpass filter. In respect to the Nyquist-Shannon sampling theorem, $f_c^{(i)}$ is constrained to strictly lower values than the Nyquist frequency.

The two parameters $K^{(i)}$ and $Q^{(i)}$ are therefore chosen to be the learnable variables in TimeScaleNet, and the five coefficients used in the difference equation can be expressed using (3) , with $\nu^{(i)} = \left[1 + K^{(i)}/Q^{(i)} + \left(K^{(i)}\right)^2\right]^{-1}$. These expressions have been obtained using a standard bilinear transformation of continuous-time, second-order bandpass filters, with frequency warping compensation [30]:

$$\begin{cases} b_i^{(0)} = \left(K^{(i)}/Q^{(i)}\right) \times \nu^{(i)} \\ b_i^{(1)} = 0 \\ b_i^{(2)} = -b_i^{(0)} \\ a_i^{(1)} = 2 \times \left[\left(K^{(i)}\right)^2 - 1\right] \times \nu^{(i)} \\ b_i^{(2)} = \left[1 - \left(K^{(i)}/Q^{(i)}\right) + \left(K^{(i)}\right)^2\right] \times \nu^{(i)} \end{cases} \quad (3)$$

In order to keep the phase information the same as in the initial waveform for each filters, we implemented a zero-phase filter using forward-backward time filtering: $x[n]$ is filtered using (1) and (3). The output is then time-reversed, filtered a second time using the same difference equation and coefficients, and time-reversed again. Using this procedure, the phase response of each learnable filters in the biquadratic RNN layer is truly zero : no matter what nonlinear phase response the IIR forward filter may have, this phase is completely canceled out by forward and backward filtering. The amplitude of the frequency response of the IIR filters, on the other hand, are squared, which allows to double the stopband attenuation in dB.

The corresponding custom RNN cell has been implemented using high order operations of the Tensorflow open source software library [31] that allow to recursively scan functions over arbitrarily long sequences and to unfold dynamically the computational graph at runtime. This implementation is compatible with a back-propagation-through-time process, in order to compute the derivative chain rule and to update the neural network parameters at each iterations of the machine learning process [33]. The expression of the custom biquadratic bidirectional RNN is fully differentiable, which allows to be compatible with the proposed machine learning approach for audio recognition, while being directly linked to standard digital audio signal processing approaches.

The i[th] output of the biquadratic RNN Layer with learnable variables $\left(K^{(i)}, Q^{i}\right)$ is still a time-domain signal which shares the same sampling frequency than the input waveform $x[n]$, and can be expressed using equation (4), where $h^{(i)}[n]$ is the inverse Z-transform of (2), defined by the coefficients $(b_i^{(0)}, b_i^{(1)}, b_i^{(2)}, a_i^{(1)}, a_i^{(2)})$ in (3). In (4), Flip$(\cdot)$ denotes the time-reversal operator :

$$s^{(i)}[n] = \text{Flip}\left(h^{(i)}[n] * \left(\text{Flip}\left(h^{(i)}[n] * x[n]\right)\right)\right) \quad (4)$$

In the following, the set of outputs $s^{(i)}[n]$ will be denoted as $\mathbf{S}_{i,n}$ – where i stands for the frequency channel index, and n for the time sample – the bold notation signifying that this is a two-dimensional tensor. $\mathbf{S}_{i,n}$ is fed to the next module in the neural network which is a deterministic module, without learnable parameters, and allows to compute a framed log-energy, in order to obtain a time-frequency-like representation.

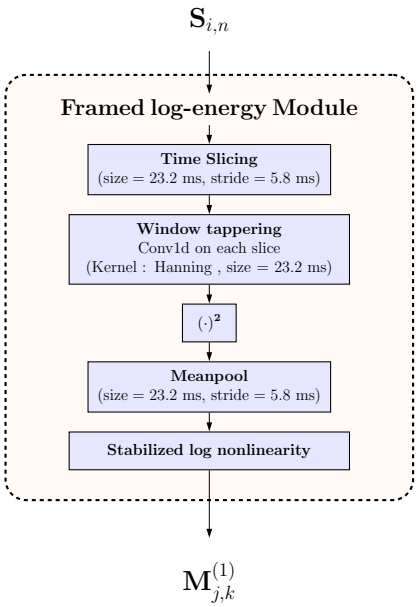


Fig. 4. (Color online) Inner architecture of the framed log-energy module, following the biquadrat ic RNN layer, and preceding the Layer Normalization Layer in BiquadNet.

As shown on Fig. 4, the framed log-energy module slices in the time domain $\mathbf{S_{i,n}}$ in order to obtain overlapped windows of 23.2 ms with a stride of 5.8 ms. These obtained frames in each frequency channels centered at the learnt frequencies $f^{(i)}$ are then multiplied with a Hanning window, squared, and averaged on each overlapping frames. This process is similar to the computation of a sliding mean quadratic value over successive overlapping timeframes in audio signal processing.

From a machine learning point of view, these successive operations correspond to a one-dimensional convolution with a kernel of width 23.2 ms, squaring, and a meanpool operation. In order to keep a lower computational cost for these deterministic operations, the one-dimensional convolution with the deterministic Hanning kernel and the meanpool operation could be replaced by a simple maxpool operation followed by rectification, as proposed in [14]. This simplification of the learnt time-frequency representation led to a weak worsening of accuracy in the classification task in our preliminary tests. We therefore chose to keep the sliding mean quadratic value computation in our implementation.

The framed log-energy representation $\mathbf{M}_{j,k}^{(1)}$ is finally computed using a stabilized logarithmic compression of each mean quadratic values, in order to produce a two-dimensional frame-level feature map. This frame-level feature map $\mathbf{M}_{j,k}^{(1)}$ – where j stands for the frequency channel index, and k for the time frame index – is intended to replace standard time-frequency representations based on short-time Fourier transforms such as mel-spectrograms, which are the most common choice of input in the majority of state-of-the-art audio classification algorithms.

This module is followed by layer normalization [34], which allows to compute layer-wise statistics and to normalize the Selu [27] nonlinear activations across all summed inputs within the layer, instead of within the batch. On contrary to batch normalization [35], [36] , whose application to RNN has been shown not to be straightforward and to lead to poor performances [37], the layer normalization approach has been shown to give promising results on RNN benchmarks, and has the great advantage of being insensitive to the mini-batch size [34].

The last layer of BiquadNet aims at achieving feature pooling across the whole frequency channels, by applying 1x1 convolutions (pointwise convolutions) followed by a Selu nonlinear activation. This kind of layer has been used for dimensionality reduction in popular computer vision approaches such as Inception [38] and its variants. In our approach, the intent of its use necessarily is not to reduce the frequency channel dimensionality, but rather to pool frequency channels together, even when the "frequency" dimension is the same as the number of filters used in the biquadratic RNN layer. In the following, this pooling property will be illustrated using experimental results, by comparing Fig. 9 and Fig. 11. For speech recognition, we think that this approach can be pertinent in order to obtain a representation that has the ability to encode well phonemes such as vowels formants and consonants, by aggregating relevant learnt frequency channels together. The output of this last layer is denoted $\mathbf{M}_{l,k}^{(2)}$ – where l stands for the pooled frequency channels index, and k for the time frame index – is then fed as the input of FrameNet, whose architecture and detailed implementation are described in the following subsection.

C. FrameNet architecture : large-scale time relationship learning on a "time-frequency-like" map

FrameNet acts at the time frame level, in order to efficiently encode the relevant relationships between time fluctuations

in different pooled frequency channels, with a large time receptive field over $\mathbf{M}_{l,k}^{(2)}$, thanks to one-dimensional atrous convolutions. Similarly to Wavenet [17], [19] architectures, we use dilation rates which are multiplied by a factor of two for each successive layers. As shown on Fig. 5, this allows to achieve a large receptive field (31 frames for a single residual subnetwork of depthwise separable atrous convolutions) with only 4 sets of one-dimensional convolutions with kernels of size 1×3. The stacked residual atrous convolutions therefore allow the network to operate on multiple time scales in the range of $[20\text{ms}; 200\text{ms}]$ without impacting too much the computational efficiency.

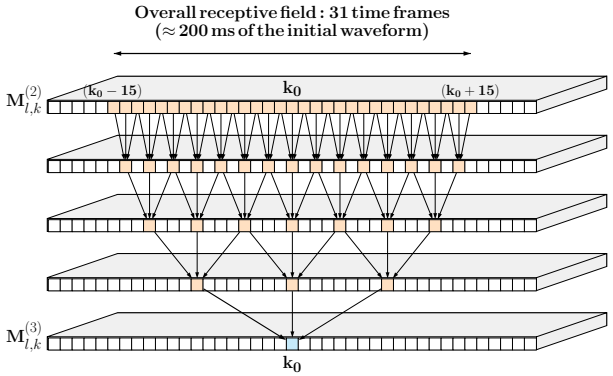


Fig. 5. (Color online) Schematics of one of the two stacks of depthwise separable atrous layers used in FrameNet, from data point of view. Each layer of this stack consists in independent convolutions for each pooled frequency channels (represented as depth on the 2D tensors of data), with only 3 nonzero coefficients. We use dilation rates which are multiplied by a factor of two for each successive layers. Only the depthwise convolution is shown here, with arrows showing the frame indexes involved in atrous convolutions for the computation of the output $\mathbf{M}_{l,k}^{(2)}$ at frame index k_0.

In our approach, we use non-causal depthwise separable convolutions, which present the considerable advantage of making a much more efficient use of the parameters available for representation learning than standard convolutions [18]. The convolutions are performed independently over every pooled channel (depthwise separable convolutions). This approach has been motivated by preliminary analysis of the energy fluctuations in different frequency channels using classical spectrogram representations. These computed depthwise convolutions are then projected onto a new channel space for each layer using a pointwise convolution (the pointwise convolution and the residual connections are not shown on Fig. 5 for sake of readability of the scheme). From a signal processing point of view, this approach aims at pooling together the contents in the soundwave that share similar time fluctuations, in order to ease the recognition task: the pointwise convolution aims at combining the pooled frequency channels in order to enhance the expressivity of the network.

As shown on Fig. 1 and Fig. 6, two of these subnetworks are stacked, and residual connections are added between each layers of the two subnetworks, thus forming two residual networks of depthwise-separable atrous 1-D convolutions. The

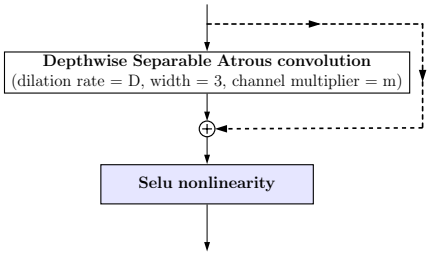


Fig. 6. (Color online) Residual connection between the successive layers of FrameNet. Each frame corresponds to a different dilation rate D, taking values 1,2,4, and 8. For the first residual network of depthwise separable atrous convolutions, the channel multiplier m is chosen to be 8, and 32 for the second one.

use of residual connections between each depthwise separable convolutional layers is intended to offer shortcut connections between layers: residual networks have been shown to offer increased representation power by circumventing some of the learning difficulties introduced by deep layers [39]. The skip connections offered by residual networks allow the information flow across the layers easier by bypassing the activations from one layer to the next. This identity mapping therefore therefore allows to prevent the saturation or deterioration of the learning process both for forward and backward computations in deep neural networks [28], [39], [40].

FrameNet shares the same ingredients as the SliceNet architecture introduced by Kaiser et al., who extensively detailed the mathematical background and the advantages of depthwise separable convolutions in [18]. In their publication, Kaiser et al. conclude that depthwise separable convolutions do not need really need atrous convolutions to be efficient for neural translation. However, our findings when developing the present TimeScaleNet architecture revealed that in our case, the use of stacked residual atrous convolutions were efficient for the intended audio recognition task, when used in conjunction with depthwise separable convolutions.

D. Training procedure

In our experiments, TimeScaleNet is trained with one-hot encoded labels, therefore allowing to compute the cross-entropy loss between estimated labels and ground truth labels. The learning and backpropagation of errors through the neural network is optimized using the Adaptive Moment Estimation (Adam) [41] algorithm, which performs an exponential moving average of the gradient and the squared gradient, and allows to control the decay rates of these moving averages. In addition to the natural decay of the learning rate that Adam performs during the learning process, we set a maximum learning rate of $\lambda_{\text{max}} = 5 \times 10^{-4}$ for the first 20 $\%$ of the total learning iterations. λ_{max} is then divided by a factor of 10 for the next 40 $\%$ of the total learning iterations, and for the remaining 40 $\%$ of the total learning iterations. The models have been implemented and

tested using the Tensorflow open source software library [31], and computations were carried out on four Nvidia GTX 1080Ti GPU cards, using mini-batches of 70 raw waveforms for spoken words recognition (resp. 120 raw waveforms for environmental sound classification) for each training steps. On this architecture, the mean computation time is only 100 ms for the whole learning process involved, for one second of audio signal (feed forward propagation, cross entropy loss, back-propagation, gradients computations, variables update using Adam). Since most of the feed-forward operations involved in TimeScaleNet could be implementable on standard audio digital signal processors, this gives us confidence that TimeScaleNet could be used for realtime inference on this kind of processors with a few adaptations, given that a considerable amount of these 100 ms are dedicated to the optimization of the learning process, which are not needed for the inference with a frozen model.

All the weights involved in layers followed with Selu activations were initialized using the He initialization [42], which relies on the idea that the variance of the weight initialization should depend on the number of inputs and outputs of the involved layer, in order to keep the variance constant from layer to layer in both the feed forward direction and back-propagation direction, which eases the learning process. The He initialization has been specifically developed for rectified linear units activations, which share some of the characteristics with the Selu activations we use in TimeScaleNet. Our experiments showed that this initialization scheme allowed to achieve a better convergence than with naive random initialization schemes.

Two types of initialization schemes were tested for the learnable parameters $K^{(i)}$ and $Q^{(i)}$ used in the biquadratic RNN layer. First, we tested clipped random initializations with minimum and maximum values corresponding to the equivalent rectangular bandwidth cochlear model introduced by Patterson [43], for central frequencies spaning from 40 Hz to $f_s/2.1$.

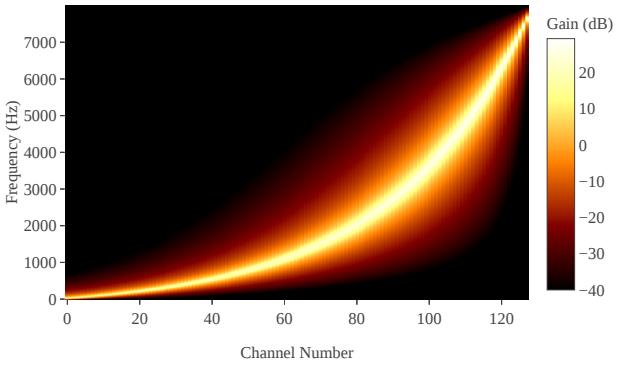


Fig. 7. (Color online) Magnitude response of the biquadratic filterbank matching the Patterson's cochlear model [43]–[45] where the bandwidth of each cochlear filter is described by an Equivalent Rectangular Bandwidth, whose parameters are chosen to match those defined by Glasberg and Moore [46].

Since this allowed a faster convergence for the model, we then chose to initalize the two learnable parameters with the values obtained using the perceptual model of critical bands introduced by Glasberg and Moore [46] (see Fig. 7). In all the studied cases, the learnt coefficients allowed to achieve significantly better classification performances than with frozen initial parameters shown on Fig. 7, therefore validating the added value of the proposed joint feature learning in the time domain achieved by BiquadNet.

III. Evaluation

A. *Datasets*

In the present paper, we evaluate the performances of the proposed TimeScaleNet for raw audio recognition, using two publicly available datasets : the Google speech commands dataset v2 [47] for speech recognition (keyword spotting) with a large dataset, and the ESC-10 dataset [48], for environmental sound classification with a rather small dataset, therefore allowing to test TimeScaleNet against overfitting problems.

The Google speech commands dataset v2 [47] consists of 105 829 utterances of 35 words recorded by 2,618 speakers, stored as one-second audio clips consisting of only one word. The audio files are encoded as 16 bits PCM / 16 kHz audio files. This dataset has recently served a competition hosted by Kaggle, which consisted in recognizing the ten words "Yes", "No", "Up", "Down", "Left", "Right", "On","Off", "Stop", and "Go" along with the "silence" class (*i.e.* no word spoken) and "unknown" class, which is randomly sampled from the remaining 25 keywords from the dataset. The dataset is split into training, validation and test sets in the ratio of 80:10:10 while making sure that the audio clips from the same person stays in the same set, using the exact procedure detailed by the maintainer of the dataset in [47].

The ESC-10 dataset [48] consists of 400 utterances of 10 types of environmental sounds, stored as five-seconds audio clips only containing one class. The 10 categories of ESC-10 are : "dog bark", "rain", "sea waves", "baby cry", "clock tick", "person sneeze", "helicopter", "chainsaw", "rooster", and "fire crackling". The audio files are encoded as 32 bits PCM / 44.1 kHz audio files. The maintainer of this dataset prearranged the files in five folds for comparable cross-validation. As a consequence, all the performance evaluations were performed using 5-fold cross-validation, using the original fold settings. In order to treat these files the exact same way than the Speech Commands dataset, we completely removed zero-valued portions at the beginning or at the end of the soundfiles, randomly cut the non-silent portions into one-second length audio files, and converted all sound files to monaural 16-bit PCM / 16 kHz audio files.

B. *Evaluation metrics*

In order to analyze precisely the performances of the proposed TimeScaleNet for the task of supervised multi-class

classification, several evaluation metrics will be used in the following. All these metrics are computed using the number of correctly recognized class examples (true positives, t_{p_i}), the number of correctly recognized examples that do not belong to the class (true negatives, t_{n_i}), and examples that either were incorrectly assigned to the class (false positives, f_{p_i}) or that were not recognized as class examples (false negatives, f_{n_i}) [49]. Using these values, for each class i of the dataset, we compute the class accuracy. The class recall R_i, which represents the effectiveness of the classifier to identify positive labels for the class i is also evaluated, along with the class precision P_i, which evaluates the class agreement of the data labels with the positive labels given by the classifier. These class-dependent metrics give more insight of the classification capabilities, and can be seen as complimentary metrics to the useful confusion matrix vizualization.

Since we achieve multi-class classification, we also compute the overall accuracy, but also the macro-averaged versions of the precision (P_M), of the recall (R_M). From R_M and P_M values, the macro-averaged F_1 score is derived, in order to evaluate the relations between data's positive labels and those given by the classifier, which allow full understanding of the overall classification task achieved by the neural network. Since the two datasets we use are relatively well balanced between classes, there is no need to evaluate micro-averaged versions of these metrics. Formulae are given in Table I for reference.

TABLE I
EVALUATION METRICS DEFINITIONS. N IS THE NUMBER OF CLASSES.

Metric	Class i	Macro-averaged
Precision	$P_i = \frac{t_{p_i}}{t_{p_i} + f_{p_i}}$	$P_M = \frac{1}{N}\sum_{i=1}^{N} P_i$
Recall	$R_i = \frac{t_{p_i}}{t_{p_i} + f_{n_i}}$	$R_M = \frac{1}{N}\sum_{i=1}^{N} R_i$
F_1 score	$F_{1\,i} = \frac{2t_{p_i}}{2t_{p_i} + f_{n_i} + f_{p_i}}$	$\frac{2P_M R_M}{P_M + R_M}$

IV. RESULTS AND DISCUSSION

In this section, we present the experiment results of sound classification for both the task of keyword recognition using the Speech Commands Dataset and the task of environmental sound classification using the ESC-10 Dataset.

For the Speech Commands Dataset, the learning process has been performed using TimeScaleNet during 45 epochs, without dropout regularization. These 45 epochs correspond to 25000 iterations, each with a batch of 70 soundfiles of 1 second. Each 50 iterations, the model was tested on the evaluation set, without updating nor computing the gradients used for learning. Using model parallelization with the four Nvidia GTX 1080Ti GPU cards, this whole process took approximately 117 hours of computation, for a total of 1200 hours of audio waveforms processed by the proposed model.

For the ESC-10 Dataset, the learning process has been performed using TimeScaleNet during 200 epochs, with dropout regularization applied to the full connected layers, with a dropout probability of 0.5. These 200 epochs correspond to 2500 iterations, each with a batch of 120 soundfiles of 1 second. Each 50 iterations, the model was tested on the evaluation fold, without updating nor computing the gradients used for learning. Using model parallelization with the four Nvidia GTX 1080Ti GPU cards, this whole process took approximately 9 hours of computation, for each fold. Since we performed a 5-fold cross-validation process for ESC-10, the whole process took approximately 45 hours of computation, for a total of 450 hours of audio waveforms processed iteratively by the proposed model.

Table II shows the obtained evaluation metrics on both the Speech Commands and the ESC-10 datasets. For the Speech Commands dataset, the mean value and standard deviation are calculated by estimating these metrics on 4 different learning processes, showing a great reproducibility. Since the ESC-10 is evaluated using a 5-fold cross-validation process, the estimation metrics are also presented with their mean value and standard deviations over the 5 experiments.

A. *Speech Commands recognition performance evaluation*

The evaluation metrics shown on Table II show that for speech commands recognition, TimeScaleNet appears to classify the 12 classes with a very high accuracy (94.87% for the evaluation set, 94.78% for the testing set, after 45 epochs of learning), with a very good homogeneity for all the classes as seen on the confusion matrix obtained for the testing set shown on Fig. 8a). The same task has also been evaluated using different configurations, including comparisons with previously published methods. The results are shown on Table III.

For reference, we first evaluated the performances of TimeScaleNet on the Speech Commands dataset with a frozen BiquadNet, using a deterministic (non-learnable) biquadratic filterbank matching the Patterson's cochlear model with Glasberg and Moore parameters, which achieved 92.4% accuracy over the testing set. A similar experiment has also been performed using a log-mel-spectrogram as an input to FrameNet, which achieved 89.7% accuracy over the testing set. For comparison purposes, this log-mel spectrogram has been computed on 128 frequency bins spanning between 40 Hz and $f_s/2.1$, and computed on overlapping Hanning-windowed frames of 23.2 ms with a stride of 5.8 ms. This parametrization allowed to build a deterministic feature map having the same dimension as the output of BiquadNet. During this comparison test, the number of parameters of FrameNet and the learning hyperparameters were kept the same than with the proposed approach. This procedure ensures a fair comparison of the proposed joint feature learning achieved by BiquadNet with a commonly

TABLE II
EVALUATION METRICS OBTAINED AFTER CONVERGENCE (45 EPOCHS OF LEARNING), FOR THE SPEECH COMMANDS DATASET [47] AND THE ENVIRONMENTAL SOUND CLASSIFICATION TASK (ESC-10), [48] USING THE PROPOSED TIMESCALENET.

Data	Cardinality	Accuracy	Precision_M	Recall_M	$F_{1,M}$
Speech Evaluation Set	4916	$\mathbf{94.87 \pm 0.24}\%$	$\mathbf{94.91 \pm 0.22}\%$	$\mathbf{94.88 \pm 0.26}\%$	$\mathbf{94.9 \pm 0.24}\%$
Speech Testing Set	5157	$\mathbf{94.78 \pm 0.26}\%$	$\mathbf{94.87 \pm 0.25}\%$	$\mathbf{94.87 \pm 0.25}\%$	$\mathbf{94.87 \pm 0.25}\%$
ESC-10, 5-fold cross-validation	364 ± 6	$\mathbf{69.71 \pm 1.91}\%$	$\mathbf{70.56 \pm 1.99}\%$	$\mathbf{69.78 \pm 1.40}\%$	$\mathbf{70.14 \pm 1.57}\%$

used handcrafted time-frequency feature representation. These two preliminary experiments mainly motivated the development of the BiquadNet part of TimeScaleNet, because this time domain approach allows to achieve a significant performance boost (over 2.5% improvement in accuracy) over handcrafted time-frequency features representations.

It is important to note that the 94.78% accuracy achieved on the testing set using the proposed TimeScaleNet matches the highest values found in [50], where the authors exhaustively benchmarked several deep learning models after careful hyperparameter tuning, for keyword spotting using the Speech Commands dataset. The different methods tested by Zhang et al. [50] are deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), convolutional recurrent neural network (CRNN) and depthwise separable convolutional neural network (DS-CNN). To the best of author's knowledge, the only published model that significantly outperforms TimeScaleNet on this particular dataset is $res15$ [51], which exhibits the best results to date with a mean accuracy of 95.8 $\%$. $res15$ shares some characteristics with the FrameNet subnet, and could be compatible with the 2D map at the output of BiquadNet. Although not being in the scope of the present paper, we intend to evaluate the performances of an approach mixing the BiquadNet approach with a subnet following the same kind of architecture than the ones proposed by Tang et al. in [51].

In order to further compare the performances of TimeScaleNet with existing methods, we performed the same keyword recognition task using the $cnn-trad-fpool3$ model proposed by Sainath *et al.* in [52]. We evaluated this CNN architecture both with a 40 MFCC map computed using the same window length and strides than those used in TimeScaleNet, and a with a 128 frequency bins log-mel spectrogram sharing the exact same characteristics as described before. The learning process has been performed during 45 epochs, and repeated 4 times in order to evaluate a standard deviation of the obtained classification accuracies. The obtained results are shown on Table III along with those obtained using $res15$ in [51], where the authors state that they applied a band-pass filter of 20 Hz / 4 kHz to the input audio before computing the 40 MFCCs. It is also interesting to note that the chosen window lengths and strides, the different learning rate schedule and the Adam optimizer used in our implementation of $cnn-trad-fpool3$'s, along with the fact that we did not filter the signals before MFFC maps computation allowed to increase the accuracy of $cnn-trad-fpool3$ by approximately 2% when compared with the reported results with the same model in [51]. Even with this improvement, the obtained results show that TimeScaleNet performs significantly better that $cnn-trad-fpool3$, which appears to be better fitted to MFCC map inputs than to log-mel spectrograms. The net difference between TimeScaleNet and $cnn-trad-fpool3$ in its best configuration is 2.25%, which is ten times larger than the standard deviation obtained on both accuracies over 4 different learning processes, validating the fact that this net difference is statistically significant.

B. Environmental sound classification performance evaluation

Motivated by the excellent results obtained with TimeScaleNet for word recognition on the Speech Commands dataset, we investigated the environmental sound classification task, using the ESC-10 dataset, in order to investigate sound classification on waveforms that did not exhibit the same kind of time fluctuations than speech, for which the TimeScaleNet has been initially thought. It is important to note that for this particular task, we did not perform any hyper-parameters optimization. The waveforms of ESC-10 have been split in 1 seconds excerpts, and downsampled to 16 kHz. The main reason behind these choices is the fact that we intend to allow a comparison between the learnt representations at the output of BiquadNet for these two particular dataset, in order to highlight the fact that BiquadNet allows to automatically build a time-frequency like representation that adapts to the particular dataset on which TimeScaleNet is trained. The particular choice of the ESC-10 has also been motivated by the fact that its small size would allow us to investigate sensitivity to overfitting problems, since there was no sign of overfitting with the Speech Commands dataset, even without dropout regularization. One another major motivation behind the use of ESC-10 dataset is the fact that the maintainer of the dataset fully documented it in order to ease reproducible comparisons across publications.

As shown on Table II, for the ESC-10 dataset, TimeScaleNet only allows to achieve environmental sound classification with a mean accuracy of 69.71% and a standard deviation of 1.91% across the five folds. This result is far from matching the best results on environmental sound classification using raw audio on the ESC-10 dataset [53]. In [53], the authors described RawNet, whose intent is also to achieve joint feature learning in the time domain, along with sound classification. Their approach allowed to achieve 85.2% of accuracy, which is

TABLE III
COMPARISON OF WORD RECOGNITION ACCURACY USING THE SPEECH COMMANDS DATASET [47] WITH DIFFERENT KINDS OF MODELS AND INPUTS

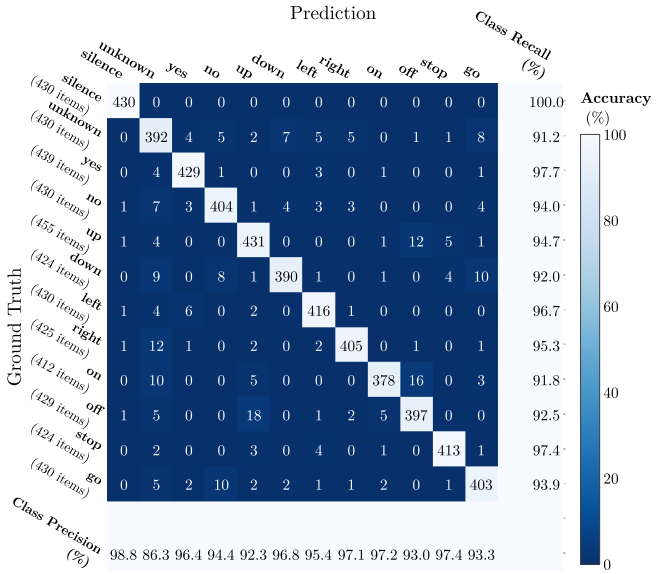
Model	Input	Accuracy
TimeScaleNet (this paper)	**Raw audio**	$\mathbf{94.87 \pm 0.24}\%$
TimeScaleNet (this paper)	Frozen BiquadNet with Patterson's cochlear model	92.4%
FrameNet (this paper)	log-mel spectrogram, 128 frequency bins	89.7%
$cnn-trad-fpool3$ [52]	40 dimensional MFCC map	$92.62 \pm 0.21\%$
$cnn-trad-fpool3$ [52]	log-mel spectrogram, , 128 frequency bins	$88.12 \pm 0.14\%$
$res15$ (data from [51])	40-dimensional MFCC map on 20 Hz / 4 kHz bandpass filtered signal [51]	$\mathbf{95.8 \pm 0.484}\%$

much better than the obtained performance of TimeScaleNet using the ESC-10 dataset, which only slightly outperforms the baseline methods proposed by the maintainer of the dataset in [48] and [54].

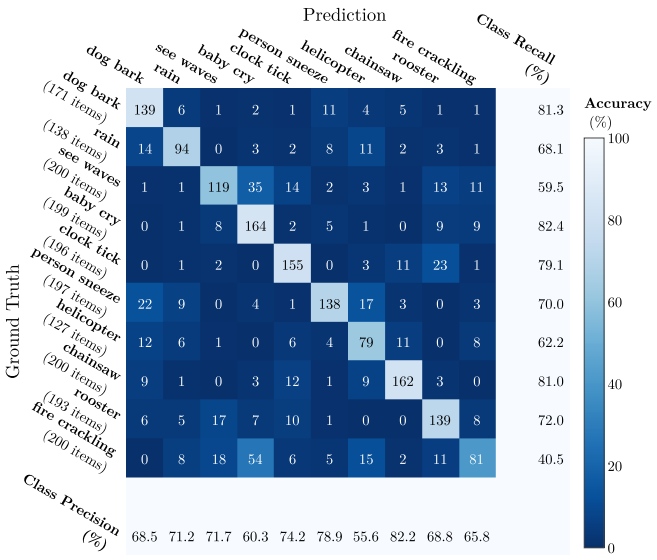
In the present paper, for comparison purposes, we deliberately chose not to change any hyperparameters for the environmental sound classification task. This may be one of the main causes of the moderate performances on this particular task. We also suspect that the rather moderate performances of TimeScaleNet for ESC could be linked to the fact that the number of parameters of TimeScaleNet are too large for such a small sized dataset. As a comparison, the number of learnt parameters used by Li *et al.* in [53] is 1.14 M, which is approximately 10 times smaller than in TimeScaleNet, for the same ESC task.

Similarly to the Speech Commands dataset, we also performed the learning process by replacing BiquadNet with a deterministic log-mel spectrogram as an input to FrameNet. The log-mel spectrogram corresponds to 128 frequency bins spanning between 40 Hz and $f_s/2.1$, computed on overlapping Hanning-windowed frames of 23.2 ms with a stride of 5.8 ms. This process allowed to achieve environmental sound classification with a mean accuracy of 71.0% and a standard deviation of 3.31% across the five folds. This result is also far from matching the accuracy obtained in [53]. This confirms that the FrameNet part of the network could be greatly improved for such a recognition task. The net difference between TimeScaleNet and FrameNet with log-mel spectrogram as input is 1.3%. However, considering the fact that the standard deviation is 2.5 times greater than this value, this difference could not be interpreted as statistically significant though, especially with such a small sized dataset.

This further confirms that the moderate performances of TimeScaleNet for ESC could be linked to the fact that FrameNet has been developed to capture time fluctuations in timescales that are commonly found in speech utterances. This assumption is motivated by the analysis of the cumulative confusion matrix obtained for the 5 cross-validations involved in the evaluation process of ESC-10 classification. As shown on Fig. 8b, the classes with the smallest recall are "sea waves", "helicopter", and "firecrackling", which are rather stationary sounds. Interestingly, previously published works on efficient environmental sound classification methods have shown that



(a) Testing set, Speech Commands



(b) Cumulative results, 5-fold cross validation ESC-10

Fig. 8. (Color online) Confusion matrix for the proposed neural network on the (a) testing set (5157 items) of the Speech Commands Dataset [47] and (b) the cumulative results of the 5-fold cross-validation of the ESC-10 dataset [48] (1821 items), after convergence ((a) : 45 epochs, (b) : 200 epochs). At the end of each row and columns, the individual class recall and precision are indicated.

convolutional network approaches show relatively poor performances for sounds with short-scale temporal structures [54], [55], but allow to better categorize stationnary sounds. This indicates that further improvements to TimeScaleNet for environmental sound classification could be achieved by modifying the FrameNet subnetwork in order to better encode stationary sounds, for which it was not intended initially.

C. Analysis of the learnt representation from raw waveforms using BiquadNet

In this subsection, we analyse the variables learnt in BiquadNet, in order to give further insight on the learning process involved. The architecture of BiquadNet has been specifically developed to automatically build a 2D map $\mathbf{M}_{l,k}^{(2)}$, that can be interpreted as an energy-like representation in 128 pooled frequency channels, with a time domain granularity of a 5.8 ms, in time frames of 23.2 ms length. As a consequence, the proposed joint feature learning process in the time domain achieved by BiquadNet allows to obtain a bi-dimensional map, which can be interpreted as a tunable time-frequency feature representation, that replaces the usual time-frequency representations commonly used as input in machine hearing.

In order to build this representation, BiquadNet first uses the previously described biquadratic RNN layer, which is directly inspired from biquadratic IIR filters used in digital signal processing. As an illustration, Fig. 9 shows the $H_{\text{dB}}^{(1)}$, which is the dB-magnitude response map of the 128 learnt filters obtained after convergence, before any nonlinearities, for the Speech Commands dataset. This representation has been obtained directly from the IIR filters expression, by computing the complex magnitude of the Z-transform of each learnt filter (see (2)), evaluated for $z = e^{j2\pi f}$ [30].

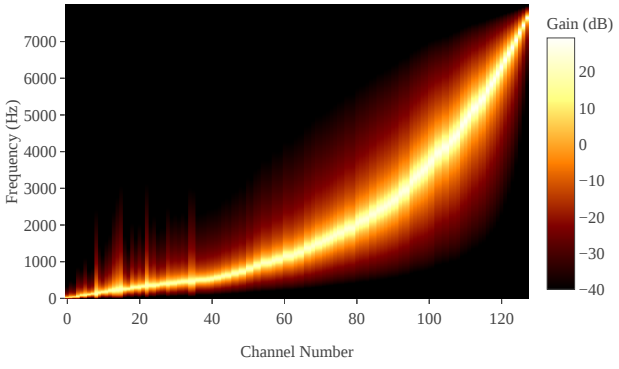
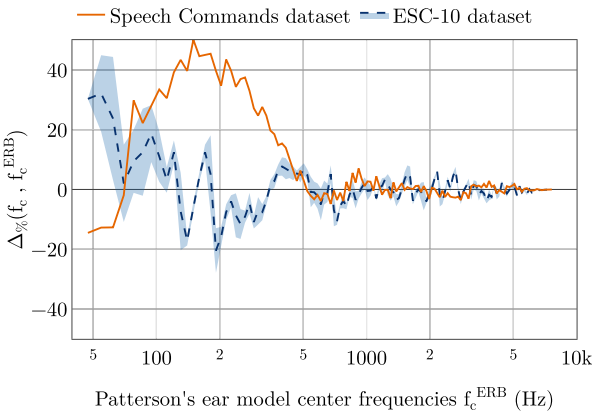
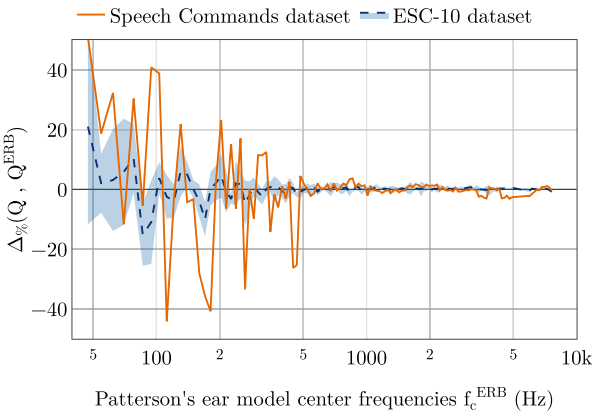


Fig. 9. (Color online) $H_{\text{dB}}^{(1)}$: Magnitude response of the learnt biquadratic filterbank before nonlinearities in the first layer of BiquadNet after convergence (45 epochs of learning), for the Speech Commands Dataset v2 [47]. The filters are sorted by ascending order of frequency at which the maximum magnitude occurs for each filters.

In order to allow a visual comparison of this learnt filterbank to the perceptual filterbank of Fig. 7, the filters on Fig. 9 are sorted by ascending order of frequency at which the maximum magnitude occurs. Although the filters share some similarities with the Patterson's cochlear model,



(a) Percentage of relative change for the central frequencies



(b) Percentage of relative change for the quality factors

Fig. 10. (Color online) Comparison of the Patterson's ear model [43]–[45] parameters defined by Glasberg and Moore [46] with the central frequency f_c^{Speech} (a) and the quality factor Q^{Speech} (b) of the learnt biquadratic filters in the first layer of BiquadNet (before nonlinearities). The values are plotted both for the Speech recognition experiment (solid line) and for the environmental sound classification experiment (dashed line : mean value for the 5 folds cross-validation, continuous shaded error bar : standard deviation).

a detailed analysis of the learnt IIR filters shows that there are some important modifications, mostly for filters having their central frequency f_c below 1 kHz. This confirms the observations made by Sainath et al. in [14], where the authors also attempted to obtain a representative filterbank, using a bank of 40, 1-dimensional convolutions of width 400 in the first stages of their neural network. As shown here, these rather large convolutions (1600 learnable parameters for 40 filters) can be replaced by an IIR approach (256 learnable parameters, for 128 filters), at the cost of using a recurrent neural network, which requires back-propagation through time for the learning process.

As an illustration, Fig. 10 shows the percentage of relative change for f_c and Q, when comparing the learnt filters and the Patterson's cochlear model. This percentage of change is simply computed using the following formula : $\Delta_{\%}(\mu, \nu) = \frac{\mu - \nu}{\nu} \times 100$, and has been computed after convergence, both for the speech recognition experiment and for the environmental sound classification experiment. Fig. 10a

shows that most of the learnt filters for speech recognition have a higher central frequency than in the perceptual model of equivalent rectangular bandwidth, thus accumulating the number of filters in the range of $[500-800Hz]$. Some of these learnt filters in this frequency range are sharper, some have a decreased quality factor. Interestingly, the particular frequency range corresponds to the typical $F1$ frequency zones of many formants of vowels in english speech [56], and could help TimeScaleNet to discriminate efficiently some phonemes present in the spoken words of the Speech Commands dataset.

When analyzing the results with ESC-10 on Fig. 10, we also observe that the learnt filters differ less from the Glasberg and Moore model than for speech recognition. Although, it is interesting to note than for the 5 folds cross-validation process, the learnt IIR filters have converged to the same kind of parameters: the standard deviation, depicted as a continuous shaded error bar, has a rather low value for frequencies above 100 Hz, which confirms that BiquadNet learns an IIR filterbank that adapts itself to the sound database automatically, rather than randomly selecting parameters for the bandpass filters. This is an interesting property, which helps explaining the excellent results obtained for speech recognition. However, potential reasons for the moderate performances obtained for environmental recognition without further optimization may be the small size of the database, or an inadapted way of encoding mid-range time dependencies using TimeScaleNet.

In order to further investigate the way BiquadNet builds a the 2D feature map $\mathbf{M}_{l,k}^{(2)}$ fed to FrameNet, we applied to $H_{\text{dB}}^{(1)}$ the mathematical operations operated by the Layer Normalization (LN) layer and the Pointwise convolution (PC) layer, along with their nonlinear activation functions. Indeed, the magnitude response shown on Fig. 9 is the strict equivalent to the output of the Framed Log-Energy Module shown on Fig. 1 and 4, that would have been obtained with a linear frequency chirp between 40 Hz and 8000 Hz taken as an input $x[n]$. This equivalence strictly stands for a linear chirp, which allows to replace the frequency axis on Fig. 9 by a timeframe number, which would give a time-frequency-like representation or the chirp $x[n]$.

This allows to compute the frequency response $H_{\text{dB}}^{(\text{BiquadNet})}$ of the equivalent (nonlinear) filterbank of the whole BiquadNet, therefore giving a higher level of interpretation of the learnt model, using the following operations :

$$H_{\text{dB}}^{(\text{BiquadNet})} = \text{Selu}\left(\text{PC}\left(\text{Selu}\left(LN\left(\text{Selu}\left(H_{\text{dB}}^{(1)}\right)\right)\right)\right)\right), \tag{5}$$

$$Selu(u) = \begin{cases} \lambda \times u & \text{if } u > 0, \\ \alpha \times (e^u - 1) & \text{if } u \leq 0 \end{cases} \tag{6}$$

$$\left(PC(U_{j,k})\right)_{l,k} = \sum_j U_{j,k} \times w_{j,l} \tag{7}$$

$$\left(LN(A_{j,k})\right)_{j,k} = \gamma_k \times \left(\frac{A_{j,k} - \mu_A}{\sigma_A}\right) + \beta_k, \tag{8}$$

where μ_A and σ_A stand for the mean and variance of A in respect to the activation values of the next layer. The values of the learnt coefficients $w_{j,l}$, γ_k and λ_k for these two layers have been extracted from the frozen model, after the 45 epochs of learning. The numerical values of α and λ used in Selu activations have been defined in [27].

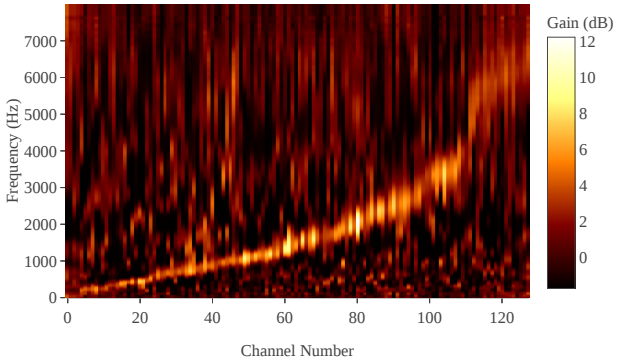


Fig. 11. (Color online) $H_{\text{dB}}^{(\text{BiquadNet})}$: Magnitude response of the equivalent filterbank at the output of BiquadNet, after convergence for the Speech Commands Dataset [47]. The filters are sorted by ascending order of frequency at which the maximum magnitude occurs for each filters.

Fig. 11 shows the computed magnitude response $H_{\text{dB}}^{(\text{BiquadNet})}$ using equations (5) to (8), for the Speech Commands Dataset. In order to ease the reading of this map, the filters were sorted by ascending order of frequency at which the maximum occurs for each filters. BiquadNet learns to build a selective filterbank which pools several frequency bands together, in order to pass them to FrameNet, which then encodes the time fluctuations in those pooled frequency bands at the frame level. Interestingly the obtained filterbank for the ESC-10 dataset does not share the same characteristics (data not shown), which supports the hypothesis that BiquadNet adapts the learnt filterbank to the dataset. Some of the channels shown on Fig. 11 exhibit frequency patterns that could be linked to vowels or nasals, whereas the last channels exhibit a frequency patterns that could serve the purpose of encoding fricatives or plosives only, with wideband, high frequency content. It is also interesting to note that the frequency at which the maximum occurs for each filters does not match the Patterson's ear model frequencies at which it has been initialized at all. The pooled frequency channels representation build by BiquadNet for speech recognition further increases the density of activations by frequencies between 200 Hz and 1000 Hz, and may explain why TimeScaleNet allows a better accuracy than with a frozen version of BiquadNet with the Patterson's cochlear model using the parameters of Glasberg and Moore.

This property is visible on Fig. 12, where the initial setting is plotted (Glasberg and Moore, between 40 Hz and 7620 Hz,

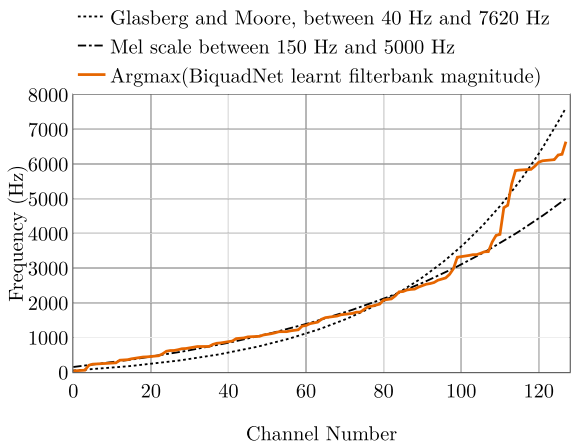


Fig. 12. (Color online) Center frequencies of the initial Patterson's cochlear model with Glasberg and Moore parameters (dotted), Frequencies at which the maximum magnitude occurs in the magnitude of $H_{\text{dB}}^{(\text{BiquadNet})}$ (solid line), and center frequencies of a 128-channels mel scale bandfilter, spanning between 150 Hz and 5000 Hz (dash-dotted line).

dotted line) together with the frequency at which the maximum occurs for each filters (solid line). The learnt maximum frequencies exhibit a linear evolution on a much larger frequency range than the Patterson's model. Interestingly, for the 100 first channels, which may mainly encode vowels and nasals, the learnt channels follow a very similar evolution than the Mel scale, which is plotted for a mel filterbank of 128 filters between 150 Hz and 5000 Hz. This is a really interesting property, since the Patterson's model and the mel scale differ greatly in the breaking frequency, and that there was initially no intent to use the mel scale in the present study. However, for the highest channel numbers depicted on Fig. 11 and 12, where the frequencies at which the maximum magnitude occurs at a larger frequency than 2500 Hz, the learnt filterbanks switches back to a Glasberg model, and clusters high frequencies together, which could help in recognizing consonants. This analysis allows to give further insight to usual handcrafted time-frequency representations used in speech recognition, and shows that there may be no best representation, since BiquadNet builds its own representation, and converges to a mix of a mel-like and a Patterson-like filterbank in the present case.

D. IIR versus FIR filtering: comparison of the proposed biquadratic RNNs with traditional CNNs for time-domain joint feature learning

In digital signal processing, filters can be designed from a given specification using either Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. As discussed earlier in the manuscript, both designs have their respective advantages and disadvantages. In machine learning, 1-D convolutional layers are the strict equivalent to FIR filterbanks. In the present paper, we developed a new kind of RNN cell, referred as biquadratic RNN, which is implemented as the strict equivalent to a tunable biquadratic, direct-form I IIR filter. In digital signal processing, when stability is ensured, IIR filters are often preferred to FIR filters because they require less computation and memory in order to perform similar filtering operations. As shown in Fig. 3, in our machine learning implementation, the Biquadratic RNN stability is ensured thanks to the range constraints on the learnable parameters $K^{(i)}$ and $Q^{(i)}$. Phase linearity is also achieved using backward-forward filtering.

In order to compare a FIR-like CNN approach to the proposed IIR-like biquadratic RNN, we implemented FIR-TimeScaleNet, which is a model that simply replaces the biquadratic RNN cells in TimeScaleNet with standard, 1-dimensional CNN cells, as proposed in [14] for time-domain joint feature learning. In order to follow Sainath et al. implementation, this convolution layer in the time domain is followed by rectification using a RELU nonlinearity. The averaging over overlapping windows [14] of 23.2 ms is performed using the exact same process as in the Framed log-energy module in BiquadNet. This process allows a fair comparison of a RNN/IIR-like approach with the CNN/FIR-like approach. As explained in [14] and [57], for a CNN approach of joint feature learning in the time domain, the kernel width used for the CNN layer is determined through extensive experimentation. This led Sainath et al. to use a kernel of width $W = 400$, which matches the value used in FIR-TimeScaleNet.

Table IV shows the computation efficiency (number of learnable parameters and number of operations for the first layer, when applied to 1 second of signal). The obtained classification accuracy on the keyword spotting task on the Speech Commands dataset [47] using the proposed TimeScaleNet and FIR-TimeScaleNet are also shown, along with the mean computation time for one iteration of the whole learning process on one second of audio. This computation time includes the feed forward propagation, cross entropy loss computation, back-propagation, gradients computations and variables updates using Adam, using four Nvidia GTX 1080Ti GPU cards and the same model parallelization on the GPU units for both models.

Since each learnable IIR filter is fully determined by only two learnable parameters in TimeScaleNet, the full number of learnable parameters in the first layer of BiquadNet is only 256. On the other hand, the FIR-like approach using CNNs involves $400 \times 128 = 51200$ parameters in the first layer, which represents 200 times more parameters to learn. The total number of operations (multiplications / addings) for a bandpass IIR implementation of a signal of length $N = 16000$ samples (1 second of signal) is $2 \times (128 \times (4+4)) \times (N+2) = 32.8 \times 10^6$ for the forward-backward biquadratic RNN implementation in TimeScaleNet. The CNN layer implemented in FIR-TimeScaleNet corresponds to $2 \times 128 \times 400 \times (N + 400 + 1) = 1.68 \times 10^9$ operations. In terms of computational cost, this is a clear win for the IIR approach, by a factor of 51, as observed in classical digital signal processing.

TABLE IV
COMPUTATION EFFICIENCY AND CLASSIFICATION ACCURACY: COMPARISON BETWEEN AN IIR AND A FIR APPROACH

Model	TimeScaleNet (IIR)	FIR-TimeScaleNet
Number of parameters (first layer)	**256**	51200
Number of operations for 1 sec. of signal	$\mathbf{32.8 \times 10^6}$	1.68×10^9
Classification accuracy	$\mathbf{94.87 \pm 0.24\%}$	$92.72 \pm 0.11\%$
Mean computation time for one learning iteration (1 sec. of signal)	105 ms	**7** ms

In order to further compare the performances of the proposed IIR-like approach with a FIR-like approach, we performed the keyword recognition task on the Speech Commands Dataset using the FIR-TimescaleNet model, whose first layer matches the one proposed by Sainath et al. in [14]. The learning process has been performed during 45 epochs, and repeated 4 times in order to evaluate a standard deviation of the obtained classification accuracies. This FIR approach allowed to obtain a classification accuracy of $92.72 \pm 0.11\%$ on the evaluation set, which is significantly lower (by a net difference of 2.15% in accuracy) than TimeScaleNet using the same data. The mean computation time is however 15 times lower for a FIR-like implementation, thanks to the optimizations for convolutional computations on GPUs. The backpropagation through time required for the IIR/RNN approach in BiquadNet is also a reason for the longer learning computation time for TimeScaleNet. This should not be a problem for realtime inference though, since forward-backward filtering using IIR filters can easily be implemented in real time, even on standard DSP units [58].

A possible reason for the lower accuracy obtained using a FIR/CNN approach could be linked to the fact that the CNN kernel width may not be well adapted for the whole audible frequency range. This kernel width is the strict equivalent to the number of taps of a FIR filterbank. However, the analysis of Figure 13 highlights the fact that, at low frequencies, a length of 400 samples for FIR filters may be insufficient to efficiently encode relevant features from raw audio at low frequencies. Figure 13 has been obtained for each of the 128 IIR learnt by BiquadNet, by calculating the number of samples of the impulse responses, whose values are higher than 0.0001 times the highest value of each impulse response. This number of samples corresponds to the length of the 128 equivalent FIR filters that would be obtained by truncating the IIR filters and discarding the smallest values of the impulse response.

Figure 13 shows that the number of coefficients proposed by Sainath et al. is big enough to efficiently encode the frequency content between 1100 Hz and 6700 Hz (corresponding to 67 filters out of the 128 filters learnt by BiquadNet). At low frequencies however, between 100 Hz and 1100 Hz, where

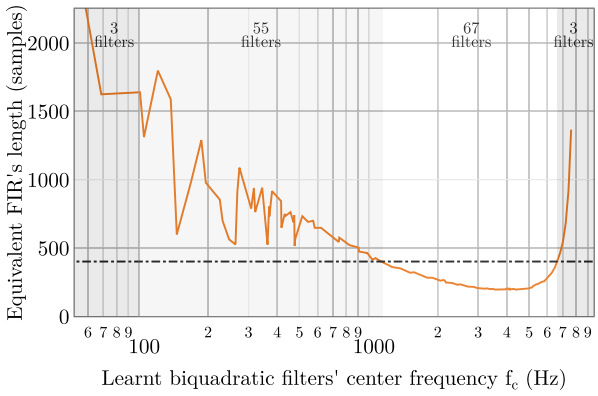


Fig. 13. (Color online) Impulse response lengths of equivalent FIR filters that would match the behavior of the learnt IIR biquadratic filters (solid line). These lengths are obtained by truncating the IIR to the portion that has values larger than 0.0001 times the highest IIR value for each filter centered at f_c. The dash-dotted line shows a length of 400, as used in [14].

BiquadNet has learnt 55 filters, the kernel width of an equivalent FIR should be much larger than 400 in order to efficiently encode the learnt perceptual filters. This result suggests that a possible improvement for a FIR/CNN approach [14] could be obtained using different kernel widths for different frequency ranges, as proposed in [59].

V. CONCLUSION

In this paper, we presented a machine learning approach of multiresolution modelling of unprocessed, time domain audio waveforms. The proposed deep neural network (TimeScaleNet) aims at merging digital signal processing techniques with new machine learning techniques, and has been specifically thought for audio recognition, with a specific intent of understanding the learning process, by justifying the network architecture from the signal point of view and visualizing the learnt representations.

The network acts at two different timescales. At the sample level, we developed BiquadNet, based on a new form of recurrent neural network cell, which is directly derived from biquadratic IIR filters found in digital signal processing. This learnable filterbank allows to build a relevant time-frequency like representation, which we have shown to self-adapt to the dataset, in order to optimize the recognition accuracy. At the frame level, we use residual networks of one-dimensional atrous convolutions (FrameNet), which help to model the time fluctuations at the frame level.

We show that this whole process allows to achieve speech recognition on a keyword spotting task with a very high accuracy, which matches the performances of the best models to date on the Speech Commands dataset. By analyzing the learnt parameters in BiquadNet for this particular task and by deriving the equivalent filterbank magnitudes from the frozen model after convergence, we give further interpretability of the proposed machine hearing process. We also show that on this particular task, the proposed neural network builds

a representation that both encodes the frequency content between 200 Hz and 3000 Hz with a pattern matching the mel-scale, and encodes higher frequency content with a pattern matching the Patterson's model. A comparison of the proposed RNN/IIR approach with a conventional CNN/FIR approach shows that BiquadNet is more computationally efficient. This analysis also gives further insight into the FIR length that would allow to efficiently learn features from raw audio at low frequencies. The proposed approach also allows to pool frequency bands together, which can efficiently encode nasals, vowels, fricatives, and plosives for speech recognition. These results allow to interpret the machine learning task in light of cognitive models of audition, while standing on both machine learning and digital signal processing solid basis.

However, the rather moderate performances for environmental sound recognition using a small dataset suggests the need for further improvements for this specific task, in order to minimize the number of parameters involved in learning for small datasets, and to modify the FrameNet approach in order to better handle stationary-like sounds, which occur more often in environmental recognition than in speech recognition.

REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] S. A. Alim and N. K. A. Rashid, "Some commonly used speech feature extraction algorithms," in *From Natural to Artificial Intelligence-Algorithms and Applications*. IntechOpen, 2018.

[3] X. Huang, A. Acero, H.-W. Hon, and R. Reddy, *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR Upper Saddle River, 2001, vol. 1.

[4] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[5] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 559–563.

[6] J. Dennis, H. D. Tran, and H. Li, "Spectrogram image feature for sound event classification in mismatched conditions," *IEEE signal processing letters*, vol. 18, no. 2, pp. 130–133, 2011.

[7] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1291–1303, 2017.

[8] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[9] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE signal processing letters*, vol. 24, no. 8, pp. 1208–1212, 2017.

[10] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2017*. Institute of Electrical and Electronics Engineers Inc., 2017.

[11] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6964–6968.

[12] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 421–425.

[13] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, pp. 1–14.

[16] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin *et al.*, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, 2017.

[17] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrener, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[18] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," in *International Conference on Learning Representations*, 2018, pp. 1–10.

[19] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5069–5073.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[21] J. Chung, C. Gulchere, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International Conference on Machine Learning*, 2015, pp. 2067–2075.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] L. R. Rabiner and B. Gold, *Theory and application of digital signal processing*. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975.

[24] M. Slaney, "An efficient implementation of the patterson-holdsworth auditory filter bank," *Apple Computer, Perception Group, Tech. Rep*, vol. 35, no. 8, 1993.

[25] R. F. Lyon, "Cascades of two-pole–two-zero asymmetric resonators are good models of peripheral auditory function," *The Journal of the Acoustical Society of America*, vol. 130, no. 6, pp. 3893–3904, 2011.

[26] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4624–4628.

[27] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 971–980.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] J. O. Smith, *Introduction to Digital Filters with Audio Applications*. W3K Publishing, 2007.

[30] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Pearson Education, 2014.

[31] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," *arXiv preprint arXiv:1605.08695*, 2016.

[32] L. B. Jackson, *Digital Filters and Signal Processing: With MATLAB® Exercises*. Springer Science & Business Media, 2013.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[34] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[36] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Advances in Neural Information Processing Systems*, 2017, pp. 1945–1953.

[37] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2657–2661.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
[40] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.
[41] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
[43] R. D. Patterson, J. Holdsworth, and M. Allerhand, "Auditory models as preprocessors for speech recognition," in *The Auditory Processing of Speech: from Auditory Periphery to Words*. Mouton de Gruyler, Berlin, 1992, pp. 67–89.
[44] R. D. Patterson, M. H. Allerhand, and C. Giguere, "Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform," *The Journal of the Acoustical Society of America*, vol. 98, no. 4, pp. 1890–1894, 1995.
[45] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," in *Auditory physiology and perception*. Elsevier, 1992, pp. 429–446.
[46] B. R. Glasberg and B. C. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing research*, vol. 47, no. 1-2, pp. 103–138, 1990.
[47] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.
[48] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1015–1018.
[49] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
[50] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.
[51] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5484–5488.
[52] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
[53] S. Li, Y. Yao, J. Hu, G. Liu, X. Yao, and J. Hu, "An ensemble stacked convolutional neural network model for environmental event sound recognition," *Applied Sciences*, vol. 8, no. 7, p. 1152, 2018.
[54] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.
[55] H. Zhou, Y. Song, and H. Shu, "Using deep convolutional neural network to classify urban sounds," in *Region 10 Conference, TENCON 2017-2017 IEEE*. IEEE, 2017, pp. 3089–3092.
[56] J. Hillenbrand, L. A. Getty, M. J. Clark, and K. Wheeler, "Acoustic characteristics of american english vowels," *The Journal of the Acoustical society of America*, vol. 97, no. 5, pp. 3099–3111, 1995.
[57] E. Variani, T. N. Sainath, I. Shafran, and M. Bacchiani, "Complex linear projection (clp): A discriminative approach to joint feature extraction and acoustic modeling." in *INTERSPEECH*, 2016, pp. 808–812.
[58] S. R. Powell and P. M. Chau, "A technique for realizing linear phase iir filters," *IEEE transactions on signal processing*, vol. 39, no. 11, pp. 2425–2435, 1991.
[59] B. Zhu, K. Xu, D. Wang, L. Zhang, B. Li, and Y. Peng, "Environmental sound classification based on multi-temporal resolution convolutional neural network combining with multi-level features," in *Pacific Rim Conference on Multimedia*. Springer, 2018, pp. 528–537.

Éric Bavu is Associate Professor in Acoustics and Signal Processing since 2009 at Laboratoire de Mécanique des Structures et des Systèmes Couplés (LMSSC), Conservatoire National des Arts et Métiers (Cnam), France. He was a former student of the Physics department of École Normale Supérieure de Cachan, France, between 2001 and 2005. He received a M.Sc in Acoustics, Signal Processing and Computer Science Applied to Music from Université Pierre et Marie Curie Sorbonne University (UPMC), France in 2005. He obtained in 2008 a Ph.D degree in Acoustics both from Université de Sherbrooke, Canada, and from UPMC, France. He has also been working between 2008 and 2009 as a post-doctoral fellow at Langevin Institute at École Supérieure de Physique et Chimie ParisTech (ESPCI), France. Since 2009, he supervised 4 Ph.D students. His main research interests include time domain audio signal processing for inverse problems, biological soft tissues imaging, time reversal techniques, moving acoustic sources tracking both in the subsonic and in the supersonic range, and deep learning methods in acoustics for sound localization and sound recognition.

Aro Ramamonjy holds a M.Sc in Acoustics, Signal Processing and Computer Science Applied to Music from Université Pierre et Marie Curie Sorbonne University (UPMC), France. He is currently pursuing his third year Ph.D. degree at Laboratoire de Mécanique des Structures et des Systèmes Couplés (LMSSC), Conservatoire National des Arts et Métiers (Cnam) under the supervision of Éric Bavu and Alexandre Garcia. His main research interest is in signal processing techniques for source localization, and statistical methods for source recognition, applied to counter-UAV systems using compact microphone arrays.

Hadrien Pujol holds a double M.Eng. degree in Mecatronics, Aerodynamics, and Aeroacoustics, delivered jointly by Karlsruhe Institute of Technology (KIT), Germany and École Nationale des Arts et Métiers ParisTech (ENSAM), France. He is currently pursuing his second year Ph.D. degree at Conservatoire National des Arts et Métiers (Cnam) under the supervision of Éric Bavu and Alexandre Garcia. His main research interest is in deep learning based methods for acoustic source localization using microphone arrays.

Alexandre Garcia is Full Professor in Acoustics since 1996 at Conservatoire National des Arts et Métiers (Cnam), France. Between 2005 and 2011, he was head of the Acoustics Chair at Cnam. He is now member of Laboratoire de Mécanique des Structures et des Systèmes Couplés (LMSSC). He holds a M.Sc in Acoustics from Univesité du Maine, Le Mans, France. He obtained in 1984 a Ph.D degree Univesité du Maine, Le Mans, France. He has also been working between 1985 and 1989 as research engineer at Thomson-Sintra underwater acoustics, France. Since 2005, he supervised 7 Ph.D students. His main research interests in the last few years have involved inverse problems in acoustics, 3D spatial audio reproduction, and acoustic imaging in adverse conditions.