LINEAR TIME COMPLEXITY DEEP FOURIER SCATTERING NETWORK AND EXTENSION TO NONLINEAR INVARIANTS

Randall Balestriero
Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005, USA
randallbalestriero@gmail.com

Hervé Glotin
DYNI, LSIS, Machine Learning and Bioacoustics team
AMU, University of Toulon, ENSAM, CNRS
La Garde, France
glotin@univ-tln.fr

ABSTRACT

In this paper we propose a scalable version of a state-of-the-art deterministic time-invariant feature extraction approach based on consecutive changes of basis and nonlinearities, namely, the scattering network. The first focus of the paper is to extend the scattering network to allow the use of higher order nonlinearities as well as extracting nonlinear and Fourier based statistics leading to the required invariants of any inherently structured input. In order to reach fast convolutions and to leverage the intrinsic structure of wavelets, we derive our complete model in the Fourier domain. In addition of providing fast computations, we are now able to exploit sparse matrices due to extremely high sparsity well localized in the Fourier domain. As a result, we are able to reach a true linear time complexity with inputs in the Fourier domain allowing fast and energy efficient solutions to machine learning tasks. Validation of the features and computational results will be presented through the use of these invariant coefficients to perform classification on audio recordings of bird songs captured in multiple different soundscapes. In the end, the applicability of the presented solutions to deep artificial neural networks is discussed.

1 INTRODUCTION AND SCATTERING NETWORK

1.1 BACKGROUND

Invariants are the gems of machine learning enabling key latent space representations of given inputs. Following this analogy, precious invariants shine out by being discriminative enough to detect changes in the underlying data distribution yet with bounded variations to ensure stable and robust representations. The motivation to find informative invariants as latent representations is becoming the principal focus from deep learning to signal processing communities aiming to tackle most machine learning tasks. Undoubtedly, given infinite datasets and computational power, learning invariants will lead to the fittest descriptors. However, nowadays problems do not fit this situation forcing the use of nested architectures and supercomputers to reach, in the limit, these utopian descriptors. As an alternative, the scattering network (Mallat, 2012; Bruna & Mallat, 2013; Andén & Mallat, 2014) provides a deterministic transformation of a given input signal x through a cascade of linear and nonlinear operators which do not commute. The linear operator is able via a dimensional increase to linearize the representation in the sense that x can be expressed as a linear combination of basic yet fundamental structures. This linear transformation is in practice a wavelet transform but can be generalized to any complete or over-complete change of basis. Originally, these wavelet transforms were used with an over-complete basis derived from Morlet and Gabor wavelets (Mallat, 1989). Recently, a discrete wavelet transform scheme (Mallat, 1999) and specifically a Haar transform (Chen et al., 2014) has been used instead to reduce the computational overload of the scattering transform. This framework, however, is not suited for general tasks due to poor frequency resolution of one wavelet per octave and the not continuously differentiable Haar wavelet (Graps, 1995) making it unsuitable for biological and natural waveforms detection. Following the change of basis, a k-Lipschitz nonlinear operator is applied which must also be contractive to enforce space contraction and thus bound the output variations (Mallat, 2016). The nonlinearity used in the scattering network is the complex modulus which is piecewise linear. This surjection aims to map the transformation into a subspace of smaller radius $\{|x| \mid x \in \Omega\} \subset \Omega$ where Ω is the space of studied signals. As a result, one can see these successions of transforms as a suite of expansion and contraction of a deeper and deeper signal representation in the hope to decode, detect and separate all the underlying energy sources. These layered representations however still contain the time dimension and are thus overall extremely sparse and not translation invariant. This time sensitivity motivates the aggregation of the time dimension. This led to the scattering coefficients per se which are computed through the application of the operator S applied on each previously computed representation and each frequency band. It is defined as an order one statistic, the arithmetic mean, over the time dimension leading to a time-invariant central tendency description of the layered representation of x. The resulting scattering coefficients, when used as features in machine learning tasks, led to state-of-the-art results in music genre classification (Chen & Ramadge, 2013), texture classification (Sifre & Mallat, 2013; Bruna & Mallat, 2013) and object classification (Oyallon & Mallat, 2015). In this paper, we present a modification of the standard scattering network by replacing the complex modulus nonlinearity with a quadratic nonlinearity in order to increase the SNR while allowing to compute the complete scattering network without leaving the Fourier domain, which was before necessary after each level of decomposition.

1.2 Scattering Network

We now present formally all the steps involved in the scattering network in order to clarify notations and concepts while making it as explicit as possible to easily develop our extensions. For readers more familiar with neural networks, it is first important to note that this framework can be seen as a restricted case of a Convolutional Neural Network (LeCun & Bengio, 1995) where the filters are fixed wavelets and the nonlinearity is the complex modulus as well as some topological differences such as depicted in Fig.1 . The scattering coefficients are then computed through time averaging of each representation.

1.2.1 Hierarchical Representation

By definition a scattering network can have any fixed number of layers L defined a priori. These layers are ordered in a hierarchical fashion so that the output of layer l is the input of layer $l+1$. In the following, many presented properties and definitions hold for all $l \in \{1, ..., L\}$. Each layer l uses a specific filter-bank made of band-pass filters $\psi^{(l)}_{\lambda}$ derived by scaling the mother wavelet of layer l denoted as $\psi^{(l)}_0$ in the time domain and $\hat{\psi}^{(l)}_0$ in the Fourier domain. The dilation factors are denoted by the subscript λ leading to

$$\psi^{(l)}_{\lambda} = \frac{1}{\lambda}\psi^{(l)}_0(\frac{t}{\lambda}) \iff \hat{\psi}^{(l)}_{\lambda} = \hat{\psi}^{(l)}_0(\lambda\omega) \tag{1}$$

The collection of scaling factors λ for layer l is denoted by $\Lambda^{(l)}$. The only admissibility condition that each filter must satisfy is to have zero mean:

$$\int \psi^{(l)}_{\lambda}(t)dt = 0 \iff \hat{\psi}^{(l)}_{\lambda}(0) = 0, \forall\lambda \in \Lambda^{(l)}, \tag{2}$$

which has the following equivalent sufficient condition on each mother wavelet

$$\int \psi^{(l)}_0(t)dt = 0 \iff \hat{\psi}^{(l)}_0(0) = 0. \tag{3}$$

The finite set of continuous scale factors needed to derive the filer-bank is given as a geometric progression governed by two hyper-parameters, the number of wavelets per octave Q and the number

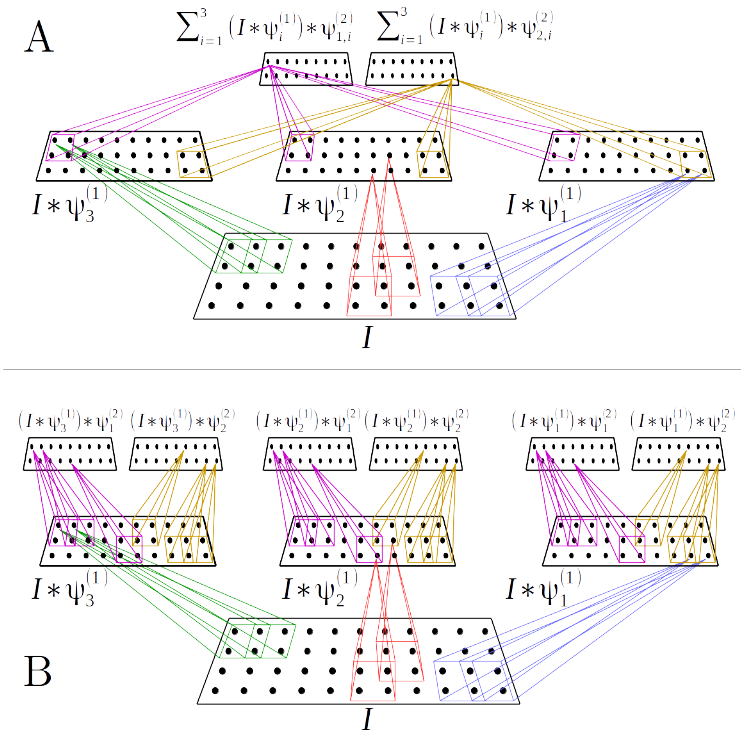
A
$\sum_{i=1}^{3}(I*\psi_i^{(1)})*\psi_{1,i}^{(2)}$
$\sum_{i=1}^{3}(I*\psi_i^{(1)})*\psi_{2,i}^{(2)}$
$I*\psi_3^{(1)}$
$I*\psi_2^{(1)}$
$I*\psi_1^{(1)}$
I
$(I*\psi_3^{(1)})*\psi_1^{(2)}$
$(I*\psi_3^{(1)})*\psi_2^{(2)}$
$(I*\psi_2^{(1)})*\psi_1^{(2)}$
$(I*\psi_2^{(1)})*\psi_2^{(2)}$
$(I*\psi_1^{(1)})*\psi_1^{(2)}$
$(I*\psi_1^{(1)})*\psi_2^{(2)}$
$I*\psi_3^{(1)}$
$I*\psi_2^{(1)}$
$I*\psi_1^{(1)}$
B
I

Figure 1: Architecture difference between the CNN (A) and the scattering network (B) without depiction of the computation of the scattering coefficients which are obtained after averaging over each obtained representation.

of octave to decompose J. The Q parameter, also called quality criteria, defines the frequency resolution, the greater it is the finer the resolution is but the more redundant will be the representation. The J parameter defines the number of octave to decompose. Since these parameters can be layer specific we now denote them as $Q^{(l)}$ and $J^{(l)}$. We thus have

$$\Lambda^{(l)} = \{2^{i/Q^{(l)}} | i = 0, ..., J^{(l)} \times Q^{(l)}\}. \tag{4}$$

When the L filter-banks are generated, it is possible to compute the L representations by iteratively applying the filter-banks and the nonlinearity. As a result, the l^{th} representation indexed by the time dimension t with the l first scales as hyper-parameters is given by:

$$\begin{gathered} X^{(0)}[x](t) := x(t) \\ X^{(1)}_{\lambda_1}[x](t) := |(X^{(0)}[x] * \psi^{(1)}_{\lambda_1})(t)|, \forall \lambda_1 \in \Lambda_1 \\ X^{(l)}_{\lambda_1,...,\lambda_l}[x](t) = |(X^{(l-1)}_{\lambda_1,...,\lambda_{l-1}}[x] * \psi^{(l)}_{\lambda_l})(t)|, \\ \forall \lambda_1 \in \Lambda^{(1)}, ..., \lambda_l \in \Lambda^{(l)} \end{gathered} \tag{5}$$

One can notice that $X^{(1)}_{\lambda_1}[x]$ coefficients form the well known wavelet transform or scalogram. We now denote by $X^{(l)}[x]$ the complete time-frequency representation for layer l if the scales are not necessary in the context.

Thinking of deeper layers as representations of more abstract concepts is inappropriate and thus not analogous to deep neural networks representations simply because deeper layer filters are not linear combination of the first layer filters. Since the used filters are renormalized to satisfy the Littlewood-Paley condition, the energy contained in each layer decays exponentially (Waldspurger, 2016). As a result, deeper layer will contain less and less energy until all events have been captured and all the next layers are zeros. With this renormalization, inverting one change of basis is instantaneous, simply add up together all the coefficients obtained from the filters application:

$$x(t) = \sum_{\lambda_1 \in \Lambda_1} (X^{(0)}[x] * \psi^{(1)}_{\lambda_1})(t). \tag{6}$$

This last property highlights one motivation of dimensional increase, event or structure separation. It is now possible to rewrite the treated signal as a combination of fundamental structures, namely, the responses of the signal with the filters linearizing the events w.r.t the new λ dimension.

1.2.2 Scattering Coefficients

For each of the L representations, one can extract the scattering coefficients by applying a scaling function $\phi^{(l)}$ on $X^{(l)}[x]$ leading to the time invariant representation $S^{(l)}[x]$. The scaling function acts as a smoothing function on a given time support and satisfies

$$\int \phi^{(l)}(t)dt = 1 \iff \hat{\phi}^{(l)}(0) = 1, \forall l. \tag{7}$$

The scaling function is usually a Gaussian filter with layer dependent standard deviations $1/\sigma^{(l)}$ in the time domain and $\sigma^{(l)}$ in the frequency domain defined as

$$\hat{\phi}^{(l)}(\omega) = e^{-\frac{\omega^2}{2\sigma^{(l)2}}}. \tag{8}$$

The greater the standard deviation is in the physical domain the more time invariant are the scattering coefficients. Ultimately, we reach global time-invariance and the scattering operator $S[x]$ reduces to an arithmetic mean over the input support. Since only the standard deviation of the scaling function is layer dependent, we denote by $\phi^{(l)}$ the l^{th} scaling function generated using $\sigma^{(l)}$. We can thus define the scattering operators as

$$\begin{gathered} S^{(0)}[x](t) := (X^{(0)}[x] * \phi^{(0)})(t), \\ S^{(1)}_{\lambda_1}[x](t) := (X^{(1)}_{\lambda_1}[x] * \phi^{(1)})(t), \forall \lambda_1 \in \Lambda^{(1)} \\ S^{(l)}_{\lambda_1,...,\lambda_l}[x](t) := (X^{(l)}_{\lambda_1,...,\lambda_l}[x] * \phi^{(l)})(t) \\ \forall \lambda_1 \in \Lambda^{(1)}, ..., \lambda_l \in \Lambda^{(l)} \end{gathered} \tag{9}$$

For machine learning tasks, using global time-invariance yield robust yet biased time invariant descriptors due to too much many-to-one possible mappings. As a result, a local or windowed scattering transform has been used (Bruna & Mallat, 2013) leading to only local time-invariance through a smaller time support for the scaling function. This tweak is possible in computer vision tasks where each input is of the same size and the role of ϕ is to bring local diffeomorphism invariance which has been shown to smooth the underlying manifold (Zoran & Weiss, 2011; 2012). However, for audio tasks and more general problems, this constant input size is rare forcing the use of ϕ for completely aggregating the time dimension.

2 Extensions

2.1 Higher Order Nonlinearity

The usual nonlinearity applied in a scattering network is the complex modulus. This nonlinearity is not everywhere differentiable but is contractive leading to an exponential decay in the energy distribution over the layers. However, as pointed out in (Waldspurger, 2016), higher order nonlinearity might be beneficial sparsity-wise and to increase the SNR. As a result, we chose to use a continuously differentiable second order nonlinearity which will have the beneficial property of adapting its contractive property for irrelevant inputs while maintaining bounded variations. This nonlinearity is defined as

$$\mathcal{P}[c] = |c|^2, \ \forall c \in \mathbb{C}. \tag{10}$$

Proof. We now prove the adaptive k-Lipschitz property of our nonlinearity

$$\begin{aligned} ||\mathcal{P}[a] - \mathcal{P}[b]|| =& || |a|^2 - |b|^2 || \\ =& ||(|a| - |b|)(|a| + |b|)|| \\ =& (|a| + |b|)|| |a| - |b| || \\ \leq& (|a| + |b|)||a - b|| \\ =& K(a, b)||a - b|| \end{aligned}$$

□

Since the input signal is renormalized so that $||x||_1 = 1$, we have that $|a| + |b| \in [0, 1[$. As a result, given the inputs constraints, $\mathcal{P}$ is a contractive operator with bounded variations. Yet, the degree of contraction will vary given the input amplitudes leading to a better SNR. This means that high amplitudes resulting from close match between the filter and the signal will be efficiently represented whereas small amplitude coefficients resulting from noise filtering and mismatches between the filter and the signal will be highly contracted. Since in practice wavelet filters catch the relevant events, this property allows high quality representations. This change will not only increase the relative sparsity of the representations but also allow us to perform some major computational tricks as describe in the following section. As a result we define the new representations as

$$\begin{gathered}
\mathcal{X}_0[x](t) := x(t) \\
\mathcal{X}^{(1)}_{\lambda_1}[x](t) := \mathcal{P}\left[(\mathcal{X}^{(0)}[x] * \psi^{(1)}_{\lambda_1})(t)\right], \forall \lambda_1 \in \Lambda^{(1)} \\
\mathcal{X}^{(l)}_{\lambda_1,...,\lambda_l}[x](t) = \mathcal{P}\left[(\mathcal{X}^{(l-1)}_{\lambda_1,...,\lambda_l}[x] * \psi^{(l)}_{\lambda_l})(t)\right], \\
\forall \lambda_1 \in \Lambda^{(1)}, ..., \lambda_l \in \Lambda^{(l)}
\end{gathered} \tag{11}$$

as well as the new scattering coefficients as

$$\begin{gathered}
\mathcal{S}^{(0)}[x](t) := (\mathcal{X}^{(0)}[x] * \phi^{(0)})(t), \\
\mathcal{S}^{(1)}_{\lambda_1}[x](t) := (\mathcal{X}^{(1)}_{\lambda_1}[x] * \phi^{(1)})(t), \forall \lambda_1 \in \Lambda^{(1)} \\
\mathcal{S}^{(l)}_{\lambda_1,...,\lambda_l}[x](t) := (\mathcal{X}^{(l)}_{\lambda_1,...,\lambda_l}[x] * \phi^{(l)})(t), \\
\forall \lambda_1 \in \Lambda^{(1)}, ..., \lambda_l \in \Lambda^{(l)}
\end{gathered} \tag{12}$$

2.2 Invariant Dispersion Coefficients

The scattering coefficients used to characterize the signal of interest are known to be efficient for stationary inputs but not descriptive enough otherwise. We thus propose to generate complementary invariant coefficients based on a dispersion measure, the variance. As a result, these complementary coefficients derived from the second order moment will help to characterize the input leading to more discriminative features while maintaining global time invariance. We now define these invariant dispersion coefficients as

$$\begin{gathered}
\mathcal{V}^{(0)}[x] := ||\mathcal{X}^{(0)}[x] - \mathcal{S}^{(0)}[x]||_2^2, \\
\mathcal{V}^{(1)}_{\lambda_1}[x] := ||\mathcal{X}^{(1)}_{\lambda_1}[x] - \mathcal{S}^{(1)}_{\lambda_1}[x]||_2^2, \forall \lambda_1 \in \Lambda^{(1)} \\
\mathcal{V}^{(l)}_{\lambda_1,...,\lambda_l}[x] := ||\mathcal{X}^{(l)}_{\lambda_1,...,\lambda_l}[x] - \mathcal{S}^{(l)}_{\lambda_1,...,\lambda_l}[x]||_2^2. \\
\forall \lambda_1 \in \Lambda^{(1)}, ..., \lambda_l \in \Lambda^{(l)}.
\end{gathered} \tag{13}$$

The resulting $\mathcal{V}^{(l)}[x]$ coefficients are thus globally time invariant whatever scaling function was used to compute $\mathcal{S}^{(l)}[x]$. In fact, these invariant dispersion coefficients represent the variance between $\mathcal{X}^{(l)}[x]$ and $\mathcal{S}^{(l)}[x]$ representations whether $\mathcal{S}^{(l)}[x]$ was globally time invariant or a smoothed version of $\mathcal{X}^{(l)}[x]$. In order for $\mathcal{V}^{(l)}[x]$ to be invariant to random permutations as well, $S^{(l)}[x]$ should be globally translation invariant and thus also globally invariant to random permutations. In addition, regarding the discriminative ability gained through the use of these second order statistic, we have that

$$\begin{gathered}
card\left(\{s \in \mathbb{L}^2(\mathbb{C}) | \mathbf{S} = (k_1, ..., k_n)\}\right) \geq \\
card\left(\{s \in \mathbb{L}^2(\mathbb{C}) | \mathbf{S} = (k_1, ..., k_n) \text{ and } \mathbf{V} = (p_1, ..., p_n)\}\right),
\end{gathered} \tag{14}$$

where $\mathbf{S}$ represents a realization of the scattering coefficients for all layers, all frequency bands, and $\mathbf{V}$ a realization of the dispersion coefficients again for all layers and all frequency bands. From this, it follows that the set of invariant coefficients $(\mathbf{S}[x], \mathbf{V}[x])$ is more discriminative leading to more precise data description than when using $(\mathbf{S}[x])$ only. The development of these presented invariant dispersion coefficients opened the door to the development of uncountably many new invariant coefficients. We now present the elaboration of the scheme in the Fourier domain and the computational tricks involved in order to reach linear complexity.

3 Fast Implementation, Fourier Domain and Linear Complexity

3.1 Introduction and Sparse Storage

One of the great benefits of the wavelet transform is the induced sparsity in the representation for certain class of signals (Elad & Aharon, 2006; Starck et al., 2010) which is seen as a quality criteria of the representation (Coifman et al., 1992). In addition of providing sparse representations, wavelets are localized filters in time and frequency domain. However, the idea to exploit this known sparsity in order to reduce the computational time of performing a transformation has not been leveraged yet. When dealing with the standard time domain, one can not know a priori where the filter with match or not the signal and thus where are the nonzeros coefficients leading to no way to have computational gains. On the other hand, applying the filter in the Fourier domain reduces to an Hadamard product and thus the resulting support is deduced from the filter support which is known to be localized. As a result, it is now possible to know a priori most of the zero coefficients positions since in the Fourier domain the filter is well localized. Also, the Fourier domain, the wavelet support is convex and compact but most importantly it can be computed a priori given the scale parameter and the mother wavelet. This motivates our choice to perform our framework including the wavelet transform, the nonlinearity $\mathcal{P}$ and the invariant features extraction in the Fourier domain leading to linear complexity. Furthermore, using the Fourier domain allows us to efficiently leverage sparse matrices leading to efficient storage and memory management on energy efficient platforms such as presented in (Esser et al., 2015). We will first present the computation of the filters in the Fourier domain as well as their convex compact support derivation. From that we present the sparse application of the filters and how to compute the nonlinearity in Fourier. Finally, we will see that extracting the invariant features can be done efficiently leading to our main result which is a linear complexity overall framework. Concerning the Fourier transform, the Danielson-Lanczos lemma (Flannery et al., 1992) will be used in order to provide a true $O(N\log(N))$ complexity for an input of size N which is a power of 2. As we will see, this requirement will always be fulfilled without any additional cost.

3.2 Sparse Fourier Filters

One particularity of the continuous wavelets such as DoG, Morlet wavelets reside in their localized compact support in the Fourier domain. In our description the used wavelet will be a Morlet wavelet but this analysis can be extended to any continuous wavelet with analytical form. We define the support of the filter $\psi_\lambda^{(l)}$ given the threshold ϵ as

$$supp_\epsilon[\psi_\lambda^{(l)}] := \{\omega | \psi_\lambda^{(l)}(\omega) > \epsilon, \omega \in [0, 2\pi]\} \tag{15}$$

As presented in (Balestriero et al., 2015) the scales define entirely the support of each wavelet. In order to develop synergistic computational tricks, we derive our framework in the Fourier domain. Let define the Morlet wavelet as

$$\hat{\psi}_{\mu_0,\sigma_0}(\omega) = H(\omega)e^{-\frac{(\omega-\mu_0)^2}{2\sigma_0^2}}, \tag{16}$$

where the parameters μ_0 and σ_0 represent respectively the center frequency and bandwidth of the mother wavelet and H is the step-wise function. The ratio between these two quantities will remain the same among all the filters, in fact, wavelets have a constant ratio of bandwidth to center frequency. These two mother hyper-parameters are taken as

$$\begin{aligned} \mu_0 &= \tfrac{\pi}{2}(2^{-1/Q}+1) \\ \sigma_0 &= \sqrt{3}(1-2^{-1/Q}). \end{aligned} \tag{17}$$

Yet, instead of using the definition of scaling as defined in section 1.2.1 we will use these two parameters as follows

$$\hat{\psi}_{\mu_0,\sigma_0}(\lambda\omega) = \hat{\psi}_{\frac{\mu_0}{\lambda},\frac{\sigma_0}{\lambda}}(\omega) := \hat{\psi}_\lambda(\omega). \tag{18}$$

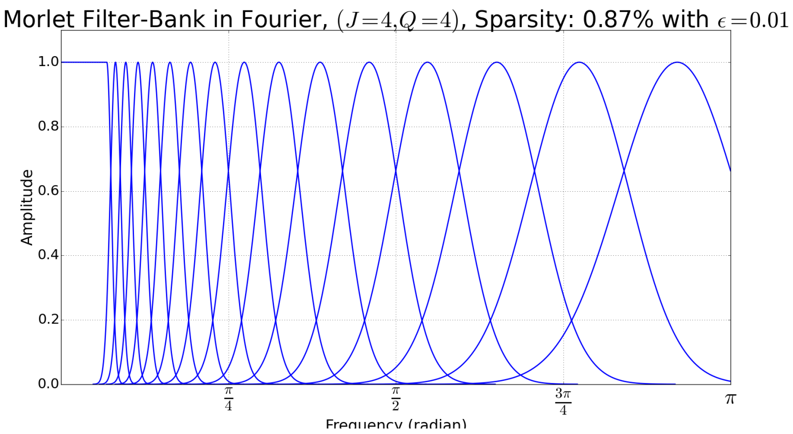

Figure 2: Filter-Bank in Fourier with 4 wavelets per octave on 4 octaves. The sparsity is of 0.94% with $\epsilon = 0.01$

Table 1: Sparsity in percentage of the Morlet Filter-Bank in Fourier for different signal sizes and realistic parameters $(J = 9, Q = 16)$.

	$\epsilon = 0.0001$	$\epsilon = 0.0000001$
$N = 524288 \ \ (2^{19})$	98.39883%	97.89803%
$N = 1048576 \ \ (2^{20})$	98.39893%	97.89812%
$N = 2097152 \ \ (2^{21})$	98.39898%	97.89816%

Table 2: Time (in sec.) needed to compute the filter-bank given the signal size with standard parameters $J = 5, Q = 16$ growing linearly with the input size on 1 CPU @1.4GHz.

Signal Size	2^{16}	2^{18}	2^{20}	2^{22}
Time in sec.	0.006	0.018	0.066	0.262

In fact, we have the following relation between the scale and the mother hyper-parameters

$$\begin{aligned}\hat{\psi}_{\mu_0,\sigma_0}(\lambda\omega) =& H(\lambda\omega)e^{-\frac{(\lambda\omega-\mu_0)^2}{2\sigma_0^2}}\\ =& H(\omega)e^{-\frac{\lambda^2(\omega-\frac{\mu_0}{\lambda})^2}{2\sigma_0^2}}\\ =& H(\omega)e^{-\frac{(\omega-\frac{\mu_0}{\lambda})^2}{2(\frac{\sigma_0}{\lambda})^2}}.\\ =& \hat{\psi}_{\frac{\mu_0}{\lambda},\frac{\sigma_0}{\lambda}}(\omega)\end{aligned}$$

Given this, we can compute explicitly the support of every filter $\psi_\lambda^{(l)}$. As one can notice these filters have a convex compact support around their center frequencies:

$$\begin{aligned}supp_\epsilon[\psi_\lambda^{(l)}] = \Big[&\tfrac{\mu_0}{\lambda} - \tfrac{\sigma_0}{\lambda}\sqrt{-2\log(\epsilon)},\\ &\tfrac{\mu_0}{\lambda} + \tfrac{\sigma_0}{\lambda}\sqrt{-2\log(\epsilon)}\Big].\end{aligned} \tag{19}$$

In Fig. 2 one can see a filter-bank example where all the filters are presented in one plot demonstrating the important sparsity inherent to wavelets. Varying the ϵ parameter affects directly the number of nonzero coefficients and we thus also present in Table 1 the exact sparsity with different input sizes and ϵ parameter. Since the support is known a priori, it is straightforward to optimize their computation and allocation through sparse vectors leading to fast filter-bank generation as presented in Table 2 with large input sizes. In fact, the input length defines the size of the generated filter since we now perform the convolution in the Fourier domain.

Concerning the $\phi^{(l)}$ filter, given its bandwidth $\sigma^{(l)}$, its support is given by:

$$supp_{\epsilon}[\phi^{(l)}] = \left[-\sigma^{(l)}\sqrt{-2\log(\epsilon)}, \sigma^{(l)}\sqrt{-2\log(\epsilon)}\right]. \tag{20}$$

Some examples are shown in Fig. 8 where different $\sigma^{(l)}$ are selected representing different time supports. For each of the filters and each of the layers, the application is done through element-wise multiplication in the Fourier domain as explained in the next section where the nonlinearity will be defined.

3.3 Nonlinearity and Filtering in Fourier

The nonlinearity used in this framework defined in section 2.1 is efficiently done in the Fourier domain through the following property

$$\mathcal{F}[|x|^2] = \mathcal{F}[x] * \mathcal{F}[x]^* \tag{21}$$

If done directly, this operation would be slower in the Fourier domain since we jump from a linear complexity to a quadratic complexity. However, one should notice from section 3.2 that we are dealing with $\mathcal{F}[x]$ which are extremely sparse but most importantly with convex compact support of size $M << N$. Exploiting this sparsity could lead to a faster convolution which would still be of quadratic complexity w.r.t the support size. However, using the convolution theorem it is possible to perform this convolution in $M\log(M)$ complexity by applying again a Fourier transform now only on the convex compact support of $\mathcal{F}[x]$. In order to have proper boundary condition and not the periodic assumption of the Fourier transform we use a zero-padded version of size$2M$ instead of M leading to exact computation of the convolution. In addition, the support size of $2M$ is the minimum required size. For a fast Fourier transform algorithm, this has to be a power of 2. As a result, in practice, the zero padding is done to reach the size which is the smallest power of 2 greater than $2M$ defined as

$$2^{\lceil \log_2(2M) \rceil}, \tag{22}$$

where $\lceil \log_2(2M) \rceil$ denotes the smallest of the greater integers. As a result in the Fourier domain we will apply another Fourier transform in order to compute this auto-correlation which will correspond to the desired nonlinearity in the time domain.

$$\mathcal{F}[|x|^2] = \mathcal{F}^{-1}\left[\mathcal{F}[\mathcal{F}[x]] \bigodot \mathcal{F}[\mathcal{F}[x]]^*\right], \tag{23}$$

where $\bigodot$ is the Hadamard product, $\mathcal{F}$ is the Discrete Fourier Transform and $\mathcal{F}^{-1}$ its inverse operator. In addition of the second Fourier transform being applied on a really small support, it is also important to note that after application of the nonlinearity the output is conjugate symmetric in the Fourier domain but since the filter-banks are always applied on $[0, \pi]$ we can store only this part for further computation and re-generate the conjugate symmetric part when applying $\phi^{(l)}$. We present this operation in the Fourier domain in Fig. 7.

In order to highlight the high sparsity encountered in the Fourier domain when dealing with this filter application, we present in Fig. 3 an example where the nonzero elements are shown. This corresponds to the first representation namely $\mathcal{X}^{(1)}[x]$. For the second representation, the input support will not be over the whole $[0, 2\pi[$ domain but around 0 and thus implies increased sparsity as demonstrated in Fig. 9. Given these two descriptions, one is able to compute $\mathcal{X}^{(l)}[x]$ for any l. We thus now present how to compute the scattering and dispersion coefficients given this representations in the Fourier domain.

3.4 Scattering Coefficients Extraction

The scattering coefficients $\mathcal{S}^{(l)}[x]$ result from the application of a Gaussian filter parameterized by its standard deviation. In the general case where global time invariance is required, this standard deviation is taken to be infinite in the time domain resulting in

$$\mathcal{S}^{(l)}_{\lambda_1,\dots,\lambda_l}[x] = \frac{1}{N}\sum_{t=1}^{N} \mathcal{X}^{(l)}_{\lambda_1,\dots,\lambda_l}[x](t). \tag{24}$$

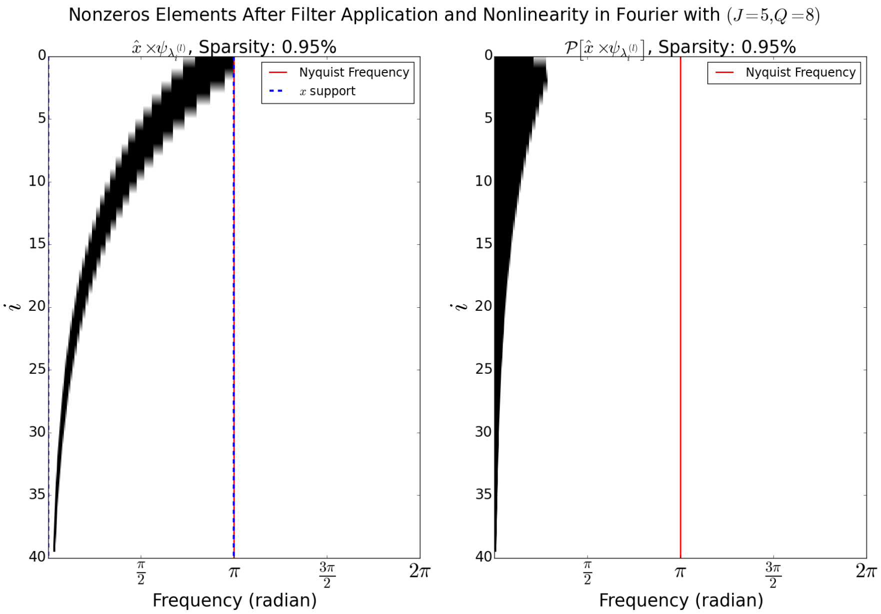

Figure 3: Nonzero elements are shown in black, after application of the filters and the nonlinearity operator $\mathcal{P}$. The x-axis corresponds to the frequency ω and the y-axis to the filter index. This follows the geometric progression of the wavelets since each line corresponds to $\psi^{(\hat{i})}\hat{x}$.

In this case, the corresponding result in Fourier is given by

$$\mathcal{S}^{(l)}[x] = \mathcal{F}\left[\mathcal{X}^{(l)}[x]\right](\omega = 0). \tag{25}$$

The invariant dispersion coefficients are extracted from the Fourier transform in a straightforward manner as shown in the Appendix which results in

$$||\mathcal{X}^{(l)}[x] - \mathcal{S}^{(l)}[x]||_2^2 = ||\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] \bigodot (1 - \mathcal{F}\left[\phi^{(l)}\right])||_2^2. \tag{26}$$

Thus $(1 - \mathcal{F}\left[\phi^{(l)}\right])$ acts as a mask to reduce the norm computation by the amount of energy captured through the scaling function application. For the case where we have global time invariance or infinite standard deviation, this mask reduces to

$$(1 - \mathcal{F}\left[\phi^{(l)}\right])(\omega) = \delta(\omega) \tag{27}$$

where δ denotes the Dirac function. As a result, the dispersion coefficients can be calculated as

$$\mathcal{V}^{(l)}[x] = 2 \sum_{\omega=1/N}^{\pi} \left(\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right](\omega)\right)^2, \tag{28}$$

which is the L2 norm computed without taking into account the coefficient at $\omega = 0$ exploiting the conjugate symmetry structure for the real input signal x. Conceptually, the $\mathcal{V}$ coefficients capture the remaining energy and thus ensures that for any depth of the scattering network, all the energy of the input is contained inside the computed invariants. In fact, one can see that $\mathcal{V}^{(l)}[x] = \sum_{i=l+1}^{\infty} \mathcal{S}^{(i)}[x]$.

3.5 Scalability

We now present some results and figures in order to highlight the high scalability of our framework with respect to the input size. Note that the current implementation is done in Python. Implementing this toolbox in C is a future work which will lead to even better results than the ones displayed below which are nevertheless already astonishing. First of all, one can see in Fig. 4 that the number of nonzero coefficients increase linearly with the input size. This result is important in nowadays

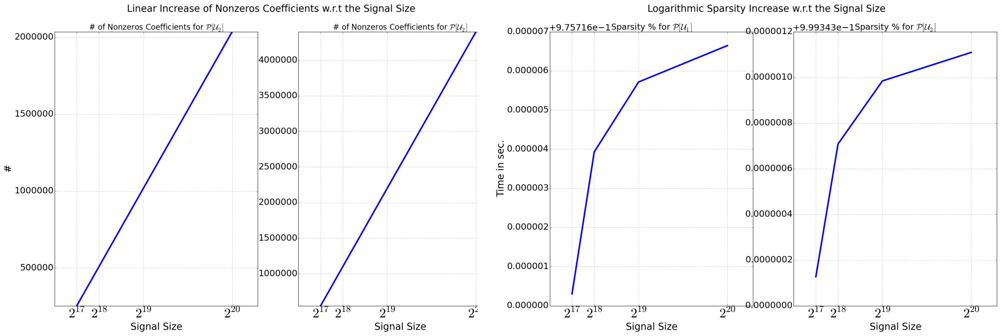

Figure 4: Left: Figure showing the increase in nonzero coefficients is linear with respect to the input size. Right: Figure showing the increase in sparsity in our representations for the two layers. The increase is logarithmic with respect to the input size.

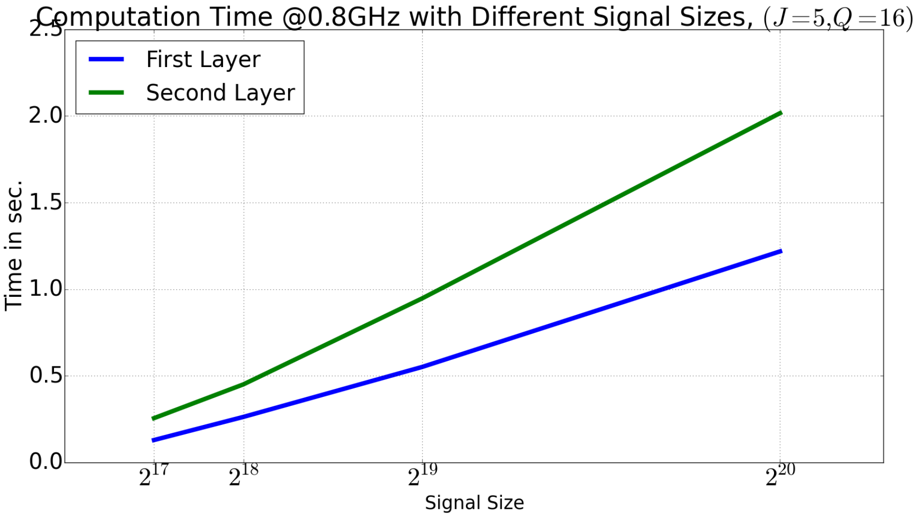

Figure 5: Needed computation time to perform the transform for the first $\mathcal{X}^{(1)}[x]$ and second $\mathcal{X}^{(2)}[x]$ layers. The need computation time is more than 16 times smaller than the known Scatnet toolbox implemented in C (`github.com/RandallBalestriero/CIGAL_GUI`) of $O(N\log(N))$ time complexity even when dealing with Fourier domain inputs. This shows the advantage of our approach which here is implemented in Python only.

paradigm where technologies allow extreme frequency sampling and thus input signals with high dimensions yet we aim to save as much memory and storage as possible. If we put this nonzero coefficients in perspective with the total possible number of coefficients we obtain our sparsity coefficient which grows logarithmically with the input size as shown in Fig. 4. This result shows the advantage of using sparse matrices which increases as the input size increases. The sparsity is thus in our case a justification to exploit the Fourier domain.

Finally, in Fig. 5 are presented some computational time for different input signals. We can see in this figure the high efficiency of our approach put in perspective of an existing C implementation of the scattering network. In fact, in this latter, one had to perform multiple inverse Fourier transforms in order to apply the nonlinearity and in order to compute the second layer for example apply again a Fourier transform and this for all the frequency bands. As a result the previously fastest known algorithm was of asymptotic complexity $O(N\log(N))$ even with a Fourier input. In addition, it did not leverage the sparsity of the representation leading to poor memory management and storing. Not however that for all the existing implementations, the complexity is linear with respect to the J and Q parameters. Finally, with our framework, one can directly store the sparse matrices of the representations leading to space saving on the hard drive in addition of the actual Random Access Memory (RAM) saving during the computation.

4 Validation on Bird Challenge

4.1 Dataset Presentation

We now validate the use of the scattering coefficients as features for signal characterization through a supervised problem of audio recordings classification. The bird song classification challenge is made of 50 classes and correspond to a small version of the BirdCLEF Challenge (Joly et al., 2015). The recordings come from the Xeno-Canto database and mainly focus on species form South America. For our task, the dataset used for training is composed of 924 files sampled at 44100 Hz. The validation set used to present our classification accuracy contains about 400 files. Computing the $\mathcal{S}[x]$ and $\mathcal{V}[x]$ features on the training and validation set takes between 2 to 3 hours depending on the set of parameters used with a 2-layer architecture on 1 CPU. The files add up to a disk usage of 4.2Go, the computed set of features however represent 450Mo. As a result, we are able to encode and extract important characteristics of the signals while reducing the amount of redundant information. We present in Fig. 101112 examples of the dataset with the waveform as well as the representation $\mathcal{X}^{(1)}_{\lambda_1}[x]$. The aim is to first demonstrate the sparsity or high SNR in the physical domain involved by using a second order nonlinearity. In addition, one can see the different frequency modulated responses that could characterize a bird song. Overall, there are some fundamental difficulties in this task. The first challenge is to deal with translation invariance. In fact, the bird songs can be captured anywhere inside each files which themselves are of many different durations, from seconds to minutes. The second difficulty resides in characterizing well enough the time-frequency patterns of each specie without being altered by the ambient noise or possible presence of other species including human voices. Finally, difficulties also arise from the machine learning point of view with large class imbalance in the dataset.

4.2 Results

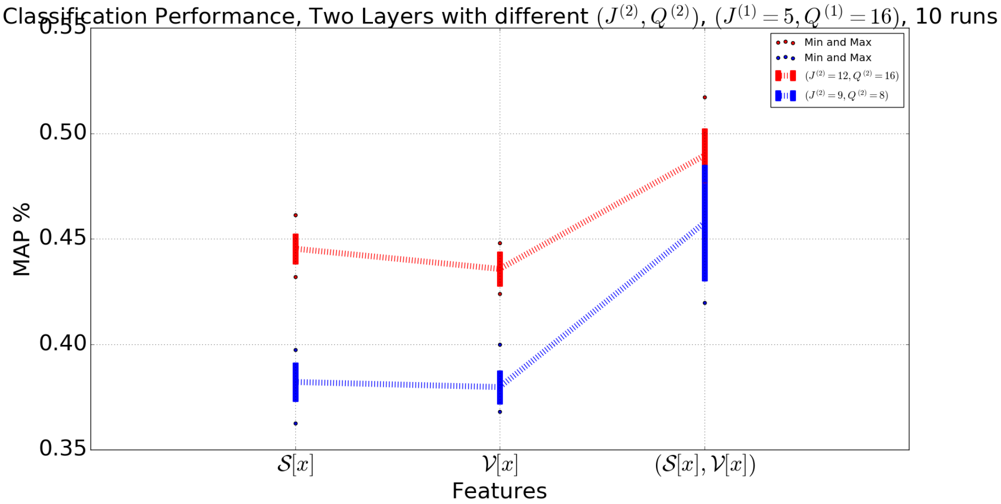

Figure 6: Classification MAP given two set of parameters for the second scattering network layer. Are also presented the results when using only the scattering coefficients $\mathcal{S}[x]$, the dispersion coefficients $\mathcal{V}[x]$ and a concatenation of both, best is 52%.

We now present the classification results obtained via our developed framework. First of all, no additional features have been engineered and thus only the features developed in the paper are used. For the classification part, we decided to use a fast algorithm in accordance with the whole scheme developed earlier and thus used random forests (Liaw & Wiener, 2002). In short, random forests are using bagging of decision trees (Breiman, 1996) and thus are able to aggregate multiple weak classifiers to end up with efficient class boundaries. One of its drawback resides in the fact that it can only create decision rule on each feature dimension without combining them as could do a logistic regression for example. In addition, we used a weighted loss function in order to deal with the imbalanced dataset (Van Hulse et al., 2007). Finally, no additional pre-processing/denoising has been used and no feature extraction/selection technique has been piped in. Yet, with this basic approach, we were able to reach an accuracy of 47.6% and a Mean Average Precision (MAP) of 52.4%. The state-of-the-art technique for this problem reached a MAP of 53% (Cha, 2016). We present in Fig.6 some accuracy results where two sets of parameters have been used for the second layer of the scattering network. In addition, we show the classification results when using each features independently and combination of the two in order to highlight their complementarity. Given the deterministic transformation used and the lack of cross-validation and fine tuning, we think of these results as promising overall while being state-of-the-art if considering solutions where no learning was involved outside of the classifier. For example, one extension on the classifier could be to use boosting algorithms (Schapire et al., 1998) or neural networks. Concerning the representation, performing cross-validation on the parameters could lead to great improvements as finally a third scattering layered could also be considered.

5 Conclusion

We presented an extension of the scattering network in order to provide more discriminative time invariant features which are complementary to each others. The derivation of a second order invariant operator as well as the use of a second order nonlinearity in the layered representation computation led to efficient characterization of audio signals opening the door of many more possible time invariant features derivation. The whole framework has been derived in the Fourier domain in order to reach linear complexity in the input size as it is now possible to compute all the layer representations and feature without leaving the Fourier domain. Sparse storage is also a milestone of our algorithm leading to not only efficient computation but smart memory management allowing this framework to be applied online on energy efficient laptops and chips. In addition, only simple arithmetic operations are used and parallel implementation can be done easily as well as GPU portage. This framework can be applied without any assumption on the input signal and thus aims to be as general as possible as a unsupervised invariant feature extraction. Finally, we hope to bring the consideration of sparse filters and Fourier based computation for deep convolutional networks. In fact, as the datasets get larger and larger, the complexity of the networks increase and convolutions might not be efficiently computed in the physical domain anymore. Since the convergence of the filter ensure their sparsity and smoothness, this consideration might help to bring deep learning to the family of scalable algorithms with the development of Fourier networks as a whole.

Acknowledgement

We thank Institut Universitaire de France for the Glotin's Chair in 'Scene Analysis'. We thank SABIOD.ORG Mission Interdisciplinaire of the CNRS and Alexis Joly for co-organisation of the LifeClef Bird Challenge. We also want to thank Mr. Romain Cosentino for his reviewing work and his help in bringing back in the physical domain some of our original sentences.

References

Bird data challenge ens paris. january 2016. URL https://challengedata.ens.fr/en/challenge/12/classify_bird_songs.html.

Joakim Andén and Stéphane Mallat. Deep scattering spectrum. *Signal Processing, IEEE Transactions on*, 62(16):4114–4128, 2014.

Randall Balestriero et al. Scattering decomposition for massive signal classification: from theory to fast algorithm and implementation with validation on international bioacoustic benchmark. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 753–761. IEEE, 2015.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1872–1886, 2013.

Xu Chen and Peter J Ramadge. Music genre classification using multiscale scattering and sparse representations. In *Information Sciences and Systems (CISS), 2013 47th Annual Conference on*, pp. 1–6. IEEE, 2013.

Xu Chen, Xiuyuan Cheng, and Stéphane Mallat. Unsupervised deep haar scattering on graphs. In *Advances in Neural Information Processing Systems*, pp. 1709–1717, 2014.

Ronald R Coifman, Yves Meyer, and Victor Wickerhauser. Wavelet analysis and signal processing. In *In Wavelets and their Applications*. Citeseer, 1992.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Steve K Esser, Rathinakumar Appuswamy, Paul Merolla, John V Arthur, and Dharmendra S Modha. Backpropagation for energy-efficient neuromorphic computing. In *Advances in Neural Information Processing Systems*, pp. 1117–1125, 2015.

Brian P Flannery, William H Press, Saul A Teukolsky, and William Vetterling. Numerical recipes in c. *Press Syndicate of the University of Cambridge, New York*, 24, 1992.

Amara Graps. An introduction to wavelets. *IEEE computational science and engineering*, 2(2): 50–61, 1995.

Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Robert Planqué, Andreas Rauber, Simone Palazzo, Bob Fisher, et al. Lifeclef 2015: multimedia life species identification challenges. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 462–483. Springer, 2015.

Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3): 18–22, 2002.

Stéphane Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

Stephane Mallat. Group invariant scattering. Communications in Pure and Applied Mathematics, vol. 65, no. 10, pp. 1331-1398, 2012.

Stéphane Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065): 20150203, 2016.

Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2865–2873, 2015.

Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, pp. 1651–1686, 1998.

Laurent Sifre and Stephane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

Jean-Luc Starck, Fionn Murtagh, and Jalal M Fadili. *Sparse image and signal processing: wavelets, curvelets, morphological diversity*. Cambridge university press, 2010.

Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pp. 935–942. ACM, 2007.

Irène Waldspurger. Exponential decay of scattering coefficients. *arXiv preprint arXiv:1605.07464*, 2016.

Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pp. 479–486. IEEE, 2011.

Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. In *Advances in Neural Information Processing Systems*, pp. 1736–1744, 2012.

A NONLINEAR INVARIANT IN THE FOURIER DOMAIN

$$
\begin{aligned}
||\mathcal{X}^{(l)}[x] - \mathcal{S}^{(l)}[x]||_2^2 = &\int \Big(\mathcal{X}^{(l)}[x](t) - \mathcal{S}^{(l)}[x](t)\Big) \\
&\Big(\mathcal{X}^{(l)}[x](t) - \mathcal{S}^{(l)}[x](t)\Big)^* dt \\
= &\int g(t)g(t)^* dt \quad g(t) = \mathcal{X}^{(l)}[x](t) - \mathcal{S}^{(l)}[x](t) \\
= &\int \mathcal{F}[g](\omega)\mathcal{F}[g^*](\omega) d\omega \;\; \text{Plancherel Theorem} \\
= &||\mathcal{F}[g]||_2^2 \\
= &||\mathcal{F}[\mathcal{X}^{(l)}[x] - \mathcal{S}^{(l)}[x]]||_2^2 \\
= &||\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] - \mathcal{F}\left[\mathcal{S}^{(l)}[x]\right]||_2^2 \quad \text{Linear Operator} \\
= &||\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] - \mathcal{F}\left[\mathcal{X}^{(l)}[x] * \phi^{(l)}\right]||_2^2 \\
= &||\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] - \\
&\mathcal{F}\left[\mathcal{F}^{-1}\left[\mathcal{F}[\mathcal{X}^{(l)}[x]] \bigodot \mathcal{F}[\phi^{(l)}]\right]\right]||_2^2 \\
= &||\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] - \mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] \bigodot \mathcal{F}\left[\phi^{(l)}\right]||_2^2 \\
= &||\mathcal{F}\left[\mathcal{X}^{(l)}[x]\right] \bigodot (1 - \mathcal{F}\left[\phi^{(l)}\right])||_2^2.
\end{aligned}
$$

B ADDITIONAL MATERIAL AND BIRD SONG REPRESENTATIONS

Using these three examples, we also present in Fig. 13 the resulting features computed on the first two layers of the scattering network in order to highlight the possibly linear hyperplanes separating these 3 species in this new feature space.

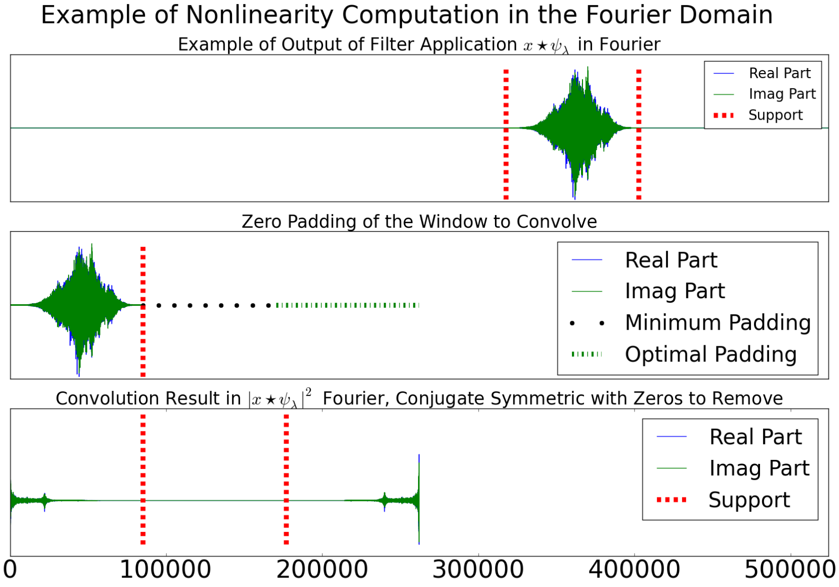

Figure 7: Demonstration of the $\mathcal{P}$ computation in the Fourier domain. Top: input after application of a specific filter. Middle: Extracted window with nonzero elements and optimal padding greater than the minimum size up to the next power of 2. Bottom: Result of the convolution done through another Fourier transform and the convolution theorem, the kept coefficients are from 0 to M since they are followed by zeros and the complex conjugate of these coefficients leading to optimal results.

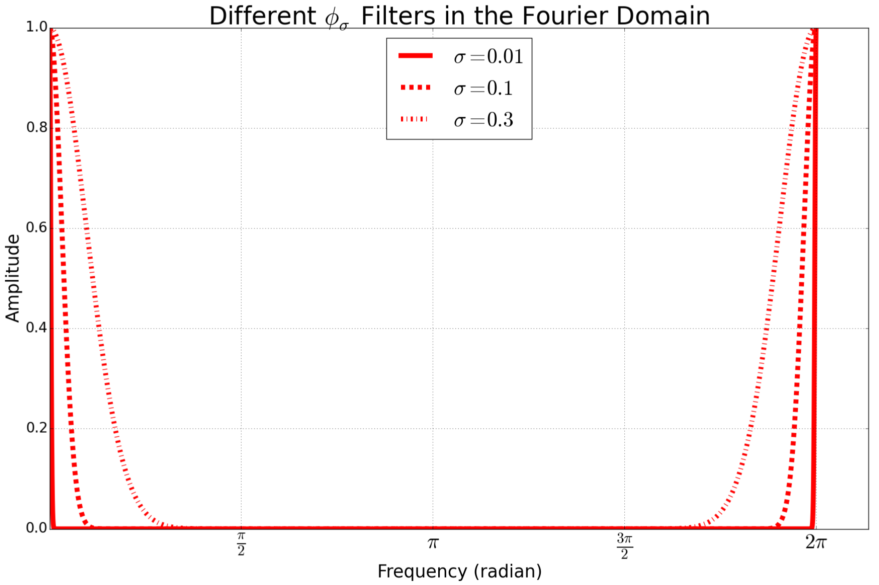

Figure 8: Some possible ϕ filters in the Fourier domain corresponding to Gaussian filtering with bandwidth in the physical domain inversely proportional to the σ in the Fourier domain without renormalization.

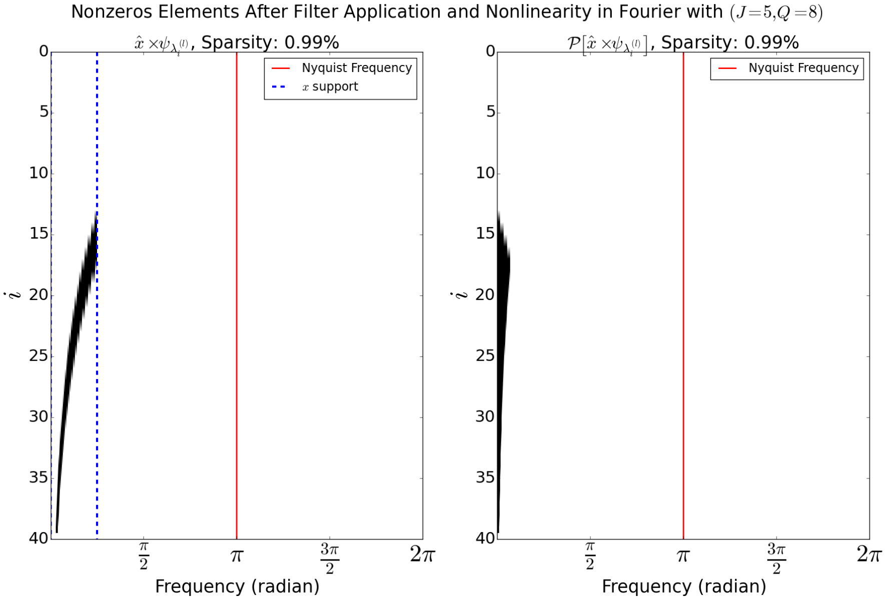

Figure 9: nonzero elements present after application of the filters and the nonlinearity operator $\mathcal{P}$ on this representation for a sparse input.

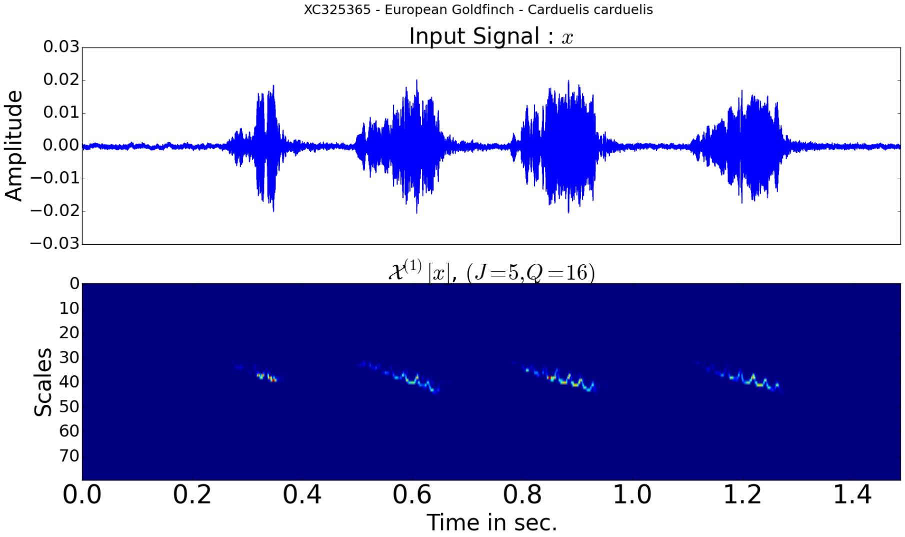

Figure 10: Example 1: transform $\mathcal{X}^{(1)}_{\lambda_1}[x]$. In this case, clear frequency modulations appear for only one source and high SNR. The noise is contracted to 0 through the nonlinearity.

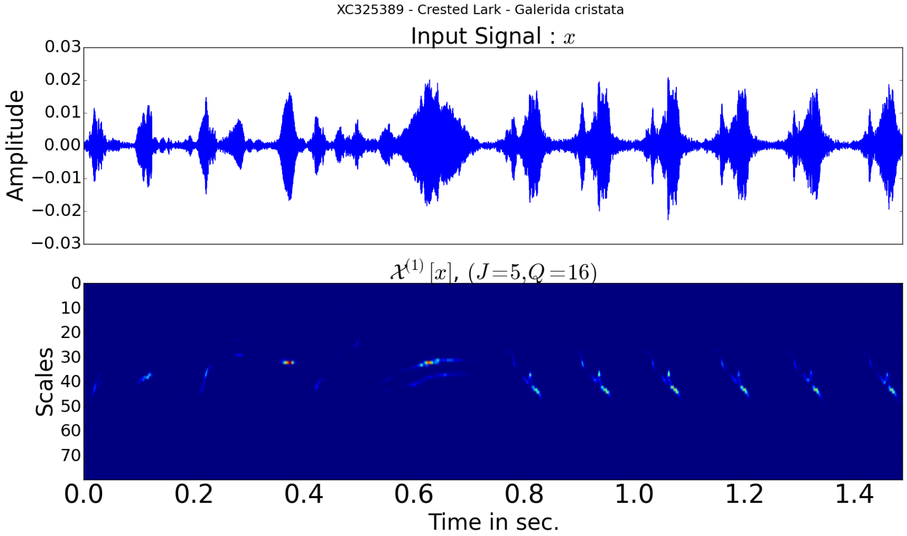

Figure 11: Example 2: transform $\mathcal{X}^{(1)}_{\lambda_1}[x]$. In this case, the source presents multiple kinds of chirps, and frequency modulated patterns. Some harmonics are detected yet it is clear that aggregation of the time dimension with this representation only will aggregate the different patterns leading to poor signal characterization leading to the need of $\mathcal{X}^{(2)}_{\lambda_1,\lambda_2}[x]$.

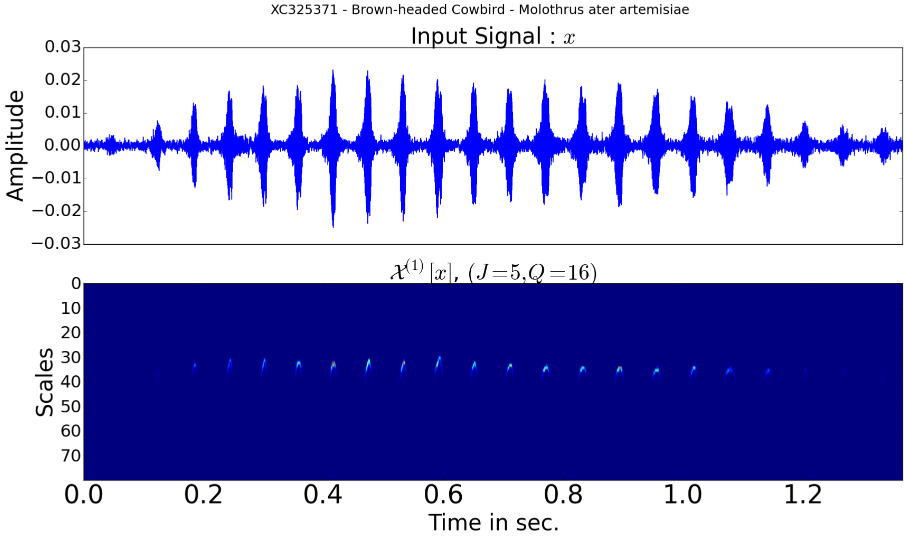

Figure 12: Example 3: transform $\mathcal{X}^{(1)}_{\lambda_1}[x]$. In this case only transients are present and almost not frequency modulation appear on the features. This kind of signal will be well captured with one layer only.

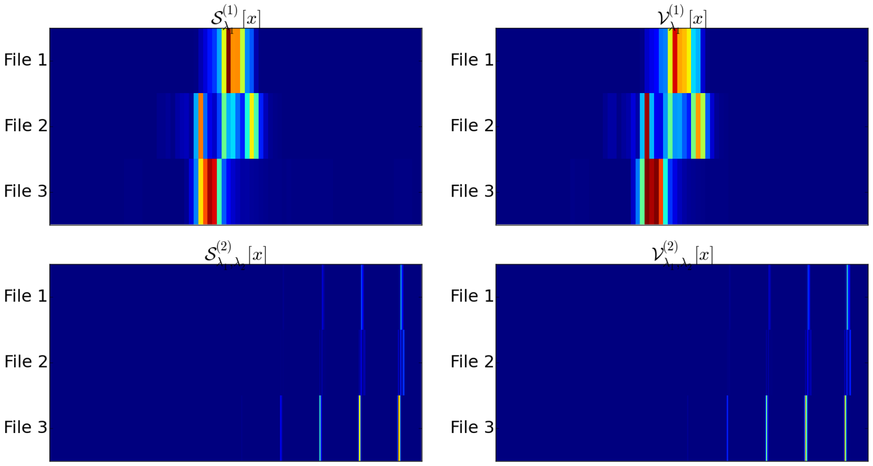

Figure 13: We present here the features extracted form the 3 examples presented in Fig. 10,11,12. The left part contains the scattering coefficients encoding the arithmetic mean whereas the right part concerns the dispersion coefficients. On the top part the features are extracted from the first layer and on the bottom are the features extracted form the second layer. It is clear that for these signals, the features of the first layer are enough to discriminate them. Notice that through global time invariance, one ends up with features vectors of exact same dimension for each signal and that they would remain the same if the input signal was translated.