Deep scattering transform and musical genre classification

Alberto Bietti
alberto.bietti@gmail.com

March, 2014

Abstract

The scattering transform defines a representation which is locally invariant to translations and stable to time-warping deformations, making well-suited for classification tasks. The transform consists of a series of wavelet transforms, complex modulus operations, and low-pass filter averaging, applied successively in a similar way to convolutional neural networks. For audio signals, scattering coefficients provide similar information to MFCCs in the first layer, while subsequent layers give additional information on larger temporal structures, such as amplitude modulation spectrum and harmonic frequency intervals [1]. We demonstrate the capabilities of the scattering representation on a musical genre classification task on the GTZAN dataset, where the goal is to assign the right genre to each sound track, in a set of 10 possible genres.

1 Introduction

For many classification problems, especially those based on natural signals such as audio or images, choosing the right representation for the data is essential in order to achieve good classification performance. Many successful approaches rely on using a good representation and feeding it into a simple linear classifier, such as a linear Support Vector Machine (SVM). These representations are typically based on local descriptors of the signal, such as SIFT or HOG descriptors for images, and MFCCs (Mel-frequency cepstral coefficients) for audio. Recently, models based on deep architectures such as convolutional neural networks have shown to provide state-of-the-art results for various classification tasks [3, 8, 5]. These methods are based on a hierarchy of several convolution filters which are learned from data, along with non-linearities and pooling operations in order to give a representation which is invariant to signal transformations. Despite the success of these methods, the underlying mathematical reasons for their success are still unknown.

The scattering transform [7] uses multiple layers of wavelet transforms, along with complex modulus non-linearities and low-pass filter averaging to provide a representation which is invariant to translation and stable to small deformations. For audio signals, the first layer captures local structures, typically in windows of the order of 25ms, giving a representation similar to MFCCs, while the second layer gives additional information on longer time intervals, thus characterizing more transient structures, such as attacks or amplitude modulation [1].

Musical genre classification tries to automatically recognize the musical genre (classical, jazz, rock, blues, etc.) of some music given an audio signal. Tzanetakis and Cook [9] use three separate sets of features in order to characterize the timbral texture, pitch content and rhythmic content of each track and help discriminate between the different genres. Most other successful methods for this

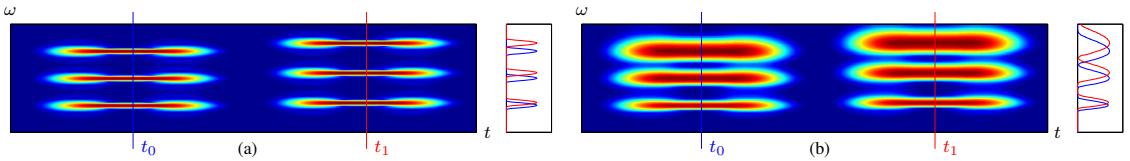


Figure 1: Spectrogram $\log|\hat{x}(t,\omega)|$ (a) and mel-frequency spectrogram $\log Mx(t,\omega)$ (b) of a harmonic signal $x(t)$ followed by a its deformation $x_\tau(t) = x((1-\epsilon)t)$ by a small dilation $\tau(t) = \epsilon t$. The graphs to the right of each spectrogram give the values of the spectrograms along the frequency axis for a fixed time in the original signal (blue) and the deformed signal (red). Source: [1]

task are based on similar representations which comprise a large variety of features, each targeted at a different aspect of the musical content [4]. Andén and Mallat [1] show that a representation given by the scattering transform alone achieves state-of-the-art results in musical genre classification on the GTZAN dataset.

We will start by motivating and describing the scattering transform and some its properties, based on [1], then we will describe experiments in musical genre classification on the GTZAN dataset.

2 Scattering transform

2.1 Invariance and stability

A good representation for classification should keep signals close to each other when they are translated versions of each other, and when they are slightly deformed. In terms of audio signals, translation invariance means that if a sound pattern starts a little later, then this delay shouldn't change the representation, while stability to time-warping deformations implies in particular that a small change in the spectrum of the signal only changes the representation slightly, thus allowing a simple linear classifier to treat these deformed signals the same way.

Formally, we seek a representation $\Phi(x)$ of a signal x which is translation invariant, i.e. if $x(t)$ and $x_c(t) := x(t-c)$ then $\Phi(x) = \Phi(x_c)$, and satisfies the following Lipschitz continuity condition: there exists $C > 0$ such that for any time-warping deformation $x_\tau(t) = x(t-\tau(t))$, with $\sup_t |\tau'(t)| < 1$, we have:

$$\|\Phi(x) - \Phi(x_\tau)\| \leq C \sup_t |\tau'(t)| \|x\|. \tag{1}$$

The modulus of the Fourier transform is translation invariant, since $|\hat{x}_c(\omega)| = |e^{-ic\omega}\hat{x}(\omega)| = |\hat{x}(\omega)|$, and the same is true for the modulus of a short-time Fourier transform (spectrogram), as long as the translation is small compared to the duration of the window function. However, these representations are not stable to deformations, as can be seen in Figure 1a: for a harmonic signal with fundamental frequency ξ, a small deformation in time $\tau(t) = \epsilon t$, with $\epsilon \ll 1$ gives a small change of $\epsilon\xi$ in the fundamental frequency, but the energy of the nth harmonic is translated by $\epsilon n\xi$, which can become large in the high frequencies and thus leads to instability.

This instability of the spectrogram is mainly due to the fixed localization in frequency of the Fourier transform. To overcome this problem, one must adapt the localization of the transform to

the different frequency domains, with larger bandwidths at high frequencies, so that the stability holds even for high frequencies. In the case of MFCCs, this is done by averaging the spectrogram over mel-frequencies (log-frequencies), or equivalently by averaging over frequencies using mel-scale filters $\hat{\phi}_\lambda(\omega)$ centered at frequencies λ, with bandwidths proportional to λ. This results in a mel-frequency spectrogram (see Figure 1b), given by

$$Mx(t,\lambda) = \frac{1}{2\pi}\int |\hat{x}(t,\omega)|^2 |\hat{\phi}_\lambda(\omega)|^2 d\omega. \tag{2}$$

Alternatively, this stability can be achieved with a wavelet transform using a similarly defined constant-Q filter bank $\{\phi_\lambda\}_\lambda$. In fact, under some appropriate condition $\lambda \gg Q/T$, one can show (see [1]) that we have:

$$Mx(t,\lambda) \approx |x * \psi_\lambda|^2 * |\phi|^2(t), \tag{3}$$

where ϕ is the spectrogram window, which corresponds to taking the modulus of wavelet coefficients and perform some time averaging to preserve some invariance to small translations (which was lost in the wavelet transform at high frequencies).

If the support T of ϕ is too large, fine-scale audio structures such as attacks, tremolos or vibratos, which are critical for good discriminability, are lost. Thus, the window size T of MFCCs is typically chosen to be small for most applications, of the order of 20-25ms. However, having a small T ignores longer time-structures, which are often useful for increasing classification performance. Delta and delta-delta MFCCs have been introduced to include some additional temporal structure by considering differences between descriptors at small intervals, but although they empirically improve performance, they don't help recover the underlying structure. In the next sections, we will see how the scattering transform is able to deal with larger window sizes by recovering the information lost by averaging with additional transforms.

2.2 Scattering transform

We now define the scattering transform from a recursive application of a wavelet transform, complex modulus non-linearities, and low-pass filtering operations.

Choice of wavelets We will consider wavelet transforms based on constant-Q filter banks (Q is the number of wavelets per octave), which are particularly adapted to audio signals. Given a complex analytical wavelet ϕ (i.e. such that $\hat{\phi}(\omega) \approx 0$ for $\omega < 0$), we define our filters to be dilated versions of ϕ, given by

$$\hat{\phi}_\lambda(\omega) = \hat{\phi}\left(\frac{\omega}{\lambda}\right), \tag{4}$$

where $\hat{\phi}$ has a bandwidth of order Q^{-1} and is centered around the frequency 1. $\hat{\phi}_\lambda$ is then centered around λ, with frequency bandwidth λ/Q. We choose $\lambda = 2^{j/Q}$ for $j \in \mathbb{Z}$, so that we have Q wavelets per octave. For low frequencies, choosing such values of λ would give wavelets which are very localized in frequency, but this isn't useful for most audio applications, so for $\lambda < 2\pi Q/T$, we replace those by $Q-1$ equally spaced filters with constant bandwidth $(2\pi Q/T)/Q = 2\pi/Q$ (which we will also call wavelets). A standard choice of such wavelets are Morlet wavelets, which are Gaussian-modulated complex exponentials.

Wavelet transform We can compute the following wavelet transform of a signal x from the filter bank $\{\phi_\lambda\}_{\lambda\in\Lambda}$, where Λ is the set of center frequencies we consider, and a low-pass filter ϕ:

$$Wx = (x * \phi(t), x * \psi_\lambda(t))_{t\in\mathbb{R},\lambda\in\Lambda}\,. \tag{5}$$

To ensure stability of this operator, we want the following to hold, for $\alpha < 1$:

$$(1-\alpha)\|x\|^2 \leq \|Wx\|^2 \leq \|x\|^2, \tag{6}$$

where $\|x\|^2 = \int |x(t)|^2 dt$ and $\|Wx\|^2 = \|x * \phi\|^2 + \sum_{\lambda\in\Lambda} \|x * \psi_\lambda\|^2$. This condition can be achieved by choosing the filter bank and the low-pass filter in such a way that the whole frequency domain is covered (see [1]). If $\alpha = 0$, W defines a unitary transformation and is called a tight frame operator (see [6]).

Wavelet modulus The wavelet modulus operator takes the complex modulus of the wavelet transform, giving a similar representation to the mel-frequency spectrogram:

$$|W|x = (x * \phi(t), |x * \psi_\lambda(t)|)_{t\in\mathbb{R},\lambda\in\Lambda}\,. \tag{7}$$

Using the inequality $||a| - |b|| \leq |a - b|$, it is easy to show that $|W|$ is contractive and hence stable to additive noise:

$$\||W|x - |W|x'\| \leq \|Wx - Wx'\| \leq \|x - x'\|. \tag{8}$$

It can be shown that this modulus operation on analytic wavelet coefficients provides a way to partly extract the envelope of the signal (demodulation). In addition, one can show that under certain conditions, the operator can be inverted, and its inverse is continuous, even though the phase of the original signal was lost by the modulus operation [10]. This phase retrieval is generally more stable when $|W|x$ is sparse, since a zero in the signal doesn't require a phase to be recovered, thus simplifying the inversion.

Scattering transform The scattering transform consists of multiple layers of coefficients, where each layer is made of the averaging of wavelet modulus coefficients by the low-pass filter ϕ. At the order zero, we have a single coefficient given by $S_0 x(t) = x * \phi(t)$, which is close to zero for audio signals. At the first order, we have:

$$S_1 x(t, \lambda_1) = |x * \psi_{\lambda_1}| * \phi(t). \tag{9}$$

This is similar to Eq. 3 and gives a stable representation, invariant to translations which are small with respect to the bandwidth of ϕ. The information that is lost by averaging is recovered by applying a new wavelet modulus transform $|W_2|$ to each of the wavelet modulus coefficients $\{|x * \psi_{\lambda_i}|\}_{\lambda_1\in\Lambda_1}$:

$$|W_2||x * \psi_{\lambda_1}| = (|x * \psi_{\lambda_1}| * \phi(t), ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}(t)|)_{t\in\mathbb{R},\lambda_2\in\Lambda_2}\,,$$

and the next layer of scattering coefficients is then given by:

$$S_2 x(t, \lambda_1, \lambda_2) = ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi(t). \tag{10}$$

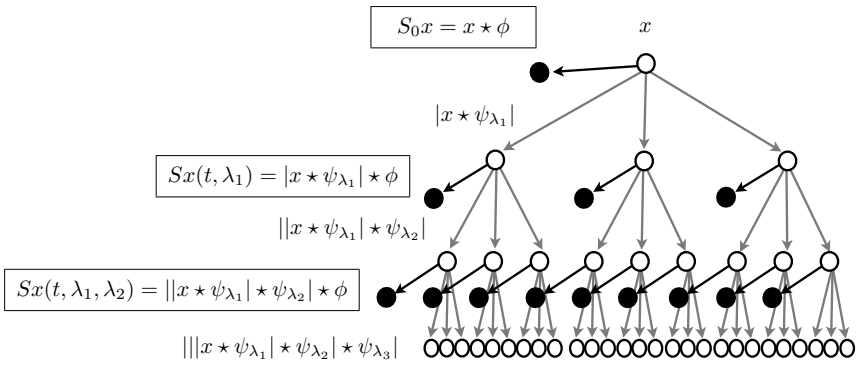


Figure 2: Hierarchical representation of scattering coefficients at multiple layers. Source: [1]

If we write

$$U_m x(t, \lambda_1, \ldots, \lambda_m) = |||x * \psi_{\lambda_1}| * \ldots| * \psi_{\lambda_m}(t)| \tag{11}$$

$$S_m x(t, \lambda_1, \ldots, \lambda_m) = |||x * \psi_{\lambda_1}| * \ldots| * \psi_{\lambda_m}| * \phi(t), \tag{12}$$

we can define scattering coefficients recursively using the following formula:

$$|W_{m+1}|U_m x = (S_m x, U_{m+1} x)\,. \tag{13}$$

Figure 2 shows the hierarchy of scattering coefficients. Thus somewhat resembles the structure of (deep) neural networks, although in the scattering transform, each layer provides some output, while the only outputs of deep networks are from the last layer. We further compare scattering networks and deep neural networks in section 2.6.

First-order coefficients give a similar representation to MFCCs, while subsequent layers capture additional structures which appear at longer time intervals. Each layer can be computed using a different set of wavelets, chosen specifically to capture the specificities of the signal at each layer. In practice, the first layer is typically taken to be a constant-Q filter bank with $Q_1 = 8$ for audio classification, since this gives a better frequency resolution and allows small differences in frequencies to be detected, while the second layer can be taken with $Q_2 = 1$ in order to capture transient structures, or $Q_2 > 1$ for some shorter structures like tremolos, which might need better frequency resolution.

2.3 Mathematical properties

If we write:

$$\|Sx\|^2 = \sum_{m=0}^{l} \|S_m x\|^2 = \sum_{m=0}^{l} \sum_{\lambda_1, \ldots, \lambda_m} \int |S_m(t, \lambda_1, \ldots, \lambda_m)|^2 dt, \tag{14}$$

then one can show that the scattering representation satisfies the Lipschitz continuity condition in Eq. 1 by using the fact that wavelets are stable to time-warping deformations [7], that is, there

exists a constant C such that:

$$\|Sx - Sx_\tau\| \leq C \sup_t |\tau'(t)| \|x\|. \tag{15}$$

Moreover, S is a contractive operator, thus stable to additive noise. If each W_m is a tight frame, then we have

$$\|U_m x\|^2 = \||W_m|U_m x\|^2 = \|S_m x\|^2 + \|U_{m+1} x\|^2, \tag{16}$$

which states that the energy at a given layer is conserved in the next layer, split across the output scattering coefficients and the nodes of the next layer: if we add a new layer to the network, it will contain all the energy that wasn't captured by the averaged scattering coefficients. Summing this for all layers $m \leq l$, we get:

$$\|x\|^2 = \|Sx\|^2 + \|U_{l+1} x\|^2. \tag{17}$$

One can show that as you go deeper in the network adding more layers, most of the energy goes into the scattering coefficients and $\|U_{l+1}\|$ goes to zero, thus we have energy conservation $\|Sx\| = \|x\|$ [7]. In practice, most of the energy is captured in the first two layers, although when T is set to larger values, the next few layers can still have a significant portion of the energy [1].

The scattering representation can be inverted by performing a deconvolution at the last layer to recover $U_l x$ from $S_l x$ with the Richardson-Lucy algorithm, and the other layers higher up in the network are then recovered by inverting each $|W_m|$, e.g. with the Griffin & Lim algorithm (see [1]).

2.4 Normalized scattering transform

The scattering transform can be normalized in order to provide additional invariance and decorrelate coefficients at different orders. The normalized coefficients $\tilde{S}$ are given by:

$$\tilde{S}_1 x(t, \lambda_1) = \frac{S_1 x(t, \lambda_1)}{\sum_{\lambda \in \Lambda_1} S_1 x(t, \lambda) + \epsilon}, \tag{18}$$

at the first order, where ϵ is a silence detection giving $\tilde{S}_1 x = 0$ for $x = 0$. The normalization makes the representation invariant to multiplication by a scalar. The next layers are normalized as follows:

$$\tilde{S}_m x(t, \lambda_1, \ldots, \lambda_{m-1}, \lambda_m) = \frac{S_m x(t, \lambda_1, \ldots, \lambda_{m-1}, \lambda_m)}{S_{m-1} x(t, \lambda_1, \ldots, \lambda_{m-1}) + \epsilon}. \tag{19}$$

Andén and Mallat [1] show that this normalization allows the detection of frequency intervals between harmonics in the second-order coefficients, and that these also characterize the amplitude modulation of the signal by giving its wavelet spectrum. Figure 3 shows scattering coefficients for different instruments, and clearly shows how the different attacks of each instrument have been detected, which are known to be crucial for discriminating different instruments.

2.5 Frequency transposition invariance

MFCCs have an additional step after computing the mel-frequency spectrogram, which is that of taking a discrete cosine transform along mel-frequencies, and only keeping a few low-frequency components. A similar transformation can be achieved by averaging the mel-frequency spectrogram along the log-frequency axis. This averaging provides some frequency transposition invariance,

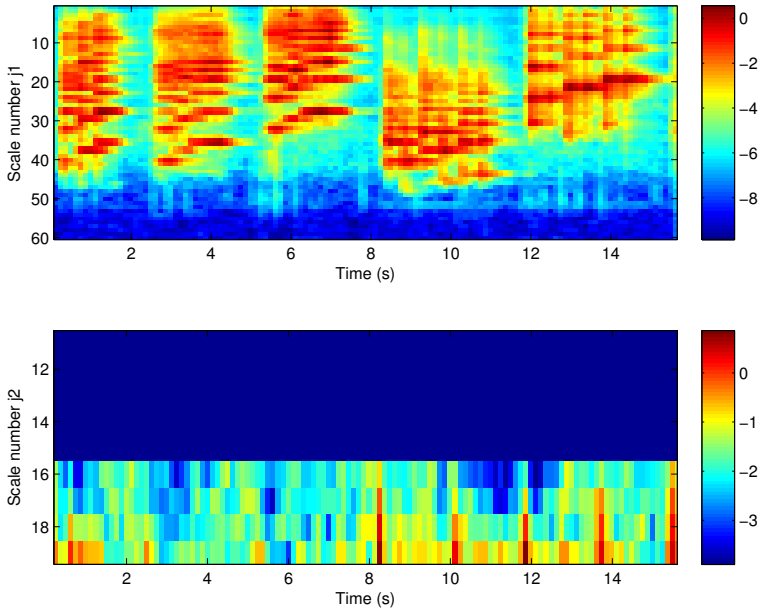


Figure 3: Scattering transform for a sequence of notes played on a piano, a cello and a violin. The leftmost sequence is a piano, the next two are cello and violin played with a bow, the last two cello and violin pizzicato. The pizzicatos have clear, short attacks, while the piano has a strong but smoother attack, and the cello and violin played with a bow have no significant attack.

which is useful for instance in speech recognition where a representation needs to be robust to changes in voice pitch. This averaging can be accomplished with another scattering transform along log-frequencies, or "quefrequencies", and as for MFCCs, the average is done on the log mel-frequency spectrogram $\log Mx(t, 2^\gamma)$, for fixed t, where γ is the quefrequency index. This scattering transform S^{fr} uses a new wavelet transform on quefrequencies, and can be directly applied to the signals output by the original scattering transform along time, for a final representation $\Phi(x) = S^{fr} \log \tilde{S}x$.

2.6 Scattering and deep neural networks

The scattering framework has striking resemblances to deep neural networks, which have been widely popular and successful on image and audio classification tasks [3, 8, 5], despite the lack of a thorough mathematical understanding.

Typical deep networks for images or audio are constructed as a series of convolutional layers (hence the name convolutional neural networks), which apply a series of convolutional filters to the signals from the previous layers. This is similar to what is done in the scattering transform. In addition, each layer in a neural network performs some non-linear operation after the linear convolution (typically a sigmoidal function, or a linear rectification unit), allowing the network

to learn non-linear representations by breaking linearity, and some layers, called pooling layers, combine the values of neighboring nodes into one by subsampling, usually with a max operation, or averaging. In scattering, the complex modulus operation acts as the non-linearity, and as a form of pooling (it merges two values into one), however there is no subsampling, and the averaging is done at output scattering coefficients, which are not propagated further in the network. Another important step in many deep neural networks is that of local contrast normalization, and this step is similar to the normalization of the scattering coefficients in the normalized scattering transform.

The main difference in the two methods is that deep neural network representations are learnt from data, usually either in an unsupervised, stage-wise fashion (called pre-training), or in a supervised manner across the whole network. Unfortunately, the learning problem in neural networks is highly non-convex, making it hard to optimize them well and not fall into local minima, but today's large datasets and recently introduced regularization techniques like dropout make this problem less desperate. In most cases, the representations learned in the first layer are quite similar to the filters used in scattering, and wavelet transforms, which suggests that no learning is needed at this stage. However, the next layers seem to learn higher-level features, which are highly dependent on the dataset used to train the network, thus giving a representation with great discriminative power.

3 Experiments

We compared the scattering transform with delta-delta-MFCCs on a musical genre classification task on the GTZAN dataset. The scattering implementation was based on the `scatnet` library[1]. The results shown in Table 1 and Table 2 were obtained on a portion of the GTZAN dataset, with 300 training tracks and 100 test tracks, with three different representations: delta-delta MFCCs with $T = 370$ms, 2-layer time scattering ($T = 370$ms), and 2-layer time scattering followed by 1-layer frequency scattering. Figure 4 shows examples scattering coefficients for tracks in different classes. We can see that these scattering representations are quite different and seem to characterize the music well, in terms of harmonic and melodic structures, rhythmic structures and temporal variations, which are the main categories of features used in [9] for musical genre classification.

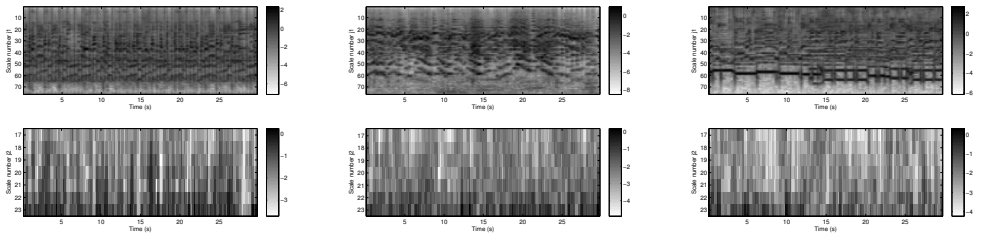
For each sound file, we consider the set of scattering coefficients at each time t sampled by the fast scattering transform algorithm (described in [1]) separately. This gives one training example at every time step of size $T/2$. All of these examples are fed into a linear SVM classifier with the class label of the corresponding sound file, and at test time, the label of a sound track is given by a majority vote on the predicted labels on each sample. A similar strategy was used for MFCCs. Using an average of the descriptors of a track instead of each sample separately gave an error rate of 25% instead of 22% for the 2-layer time-scattering, while speeding up the computations (both for training and testing) by about two orders of magnitude (a few seconds compared to several hundred seconds).

We can see in the confusion matrices in Figure 2 that time scattering corrects many of the errors made by the MFCC representation. A few errors are introduced by the time scattering that aren't made by MFCCs, but in most cases they disappear when adding frequency transposition invariance, which shows that this additional transform along the quefrequency axis reproduces the behavior of the DCT in the cepstral coefficients.

[1] Available at `http://www.di.ens.fr/data/software/scatnet/`.

Representation	Error rate
Δ-MFCC ($T = 370$ms)	33%
Time scat., $l = 2$	22%
Time scat., $l = 2$ + Freq scat.	21%

Table 1: Prediction errors on GTZAN for different representations. Results obtained with a training set of 300 tracks and a test set of 100 tracks.



(a) blues (b) classical (c) country

Figure 4: Scattering coefficients on examples of tracks from the GTZAN dataset for different classes (order two coefficients are for $j_1 = 35$).

4 Conclusion

The scattering transform gives a description of an audio signal which is stable to time-warping deformations and locally translation invariant. In addition, the multiple layers of the description capture important information about the signal, in particular on amplitude modulation spectrum and harmonic intervals in the second order coefficients, which are not obtained by standard MFCC descriptors. We saw how the scattering representation was able to outperform MFCC-based methods on GTZAN musical genre classification dataset. Similar frameworks based on scattering have been successfully applied to phone recognition, image classification, and audio and image texture discrimination [2]. This representation has striking similarities with convolutional neural networks, which have had great recent success, however, no learning is involved in scattering, which makes it more difficult to capture the specificities of a given dataset.

References

[1] Andén, J., and Mallat, S. Deep scattering spectrum. *IEEE Transactions on Signal Processing* (2011).

[2] Bruna, J. *Scattering Representations for Recognition*. 2012.

[3] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012).

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	9	0	1	0	0	2	0	0	0	0
classical	0	10	0	0	0	0	0	0	0	0
country	0	0	5	0	1	0	0	1	0	0
disco	0	0	1	5	2	0	3	1	0	0
hiphop	0	0	0	1	1	0	1	1	0	0
jazz	0	1	0	0	0	8	0	0	0	0
metal	0	0	0	0	1	0	12	0	1	0
pop	0	1	0	0	1	0	0	9	0	2
reggae	1	1	0	0	1	0	1	1	4	0
rock	0	0	2	1	1	1	0	1	0	4

(a) Δ-MFCC

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	8	0	1	0	0	2	0	0	0	1
classical	0	10	0	0	0	0	0	0	0	0
country	0	0	5	1	0	0	0	0	0	1
disco	0	0	2	9	0	0	0	1	0	0
hiphop	0	0	0	0	3	0	0	1	0	0
jazz	0	1	0	0	0	8	0	0	0	0
metal	0	0	0	0	0	0	14	0	0	0
pop	0	1	0	0	1	0	0	10	0	1
reggae	1	1	0	1	1	0	0	0	5	0
rock	0	0	2	1	0	0	0	1	0	6

(b) Scat, $l = 2$

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	9	0	1	0	0	2	0	0	0	0
classical	0	10	0	0	0	0	0	0	0	0
country	0	0	5	1	0	0	0	1	0	0
disco	0	0	2	9	0	0	0	1	0	0
hiphop	0	0	0	0	3	0	0	1	0	0
jazz	0	1	0	0	0	8	0	0	0	0
metal	0	0	0	0	0	0	14	0	0	0
pop	0	1	0	0	1	0	0	10	0	1
reggae	1	1	0	0	2	0	0	1	4	0
rock	0	0	2	0	0	0	0	1	0	7

(c) Scat, $l = 2$ + freq

Table 2: Confusion matrices on the test set for different representations. Rows are the true classes, columns the predicted classes.

[4] Lee, C.-H., Shih, J.-L., Yu, K.-M., and Lin, H.-S. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *Multimedia, IEEE Transactions on 11*, 4 (2009), 670–682.

[5] Lee, H., Pham, P., Largman, Y., and Ng, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems* (2009).

[6] Mallat, S. *A Wavelet Tour of Signal Processing*. 2008.

[7] Mallat, S. Group invariang scattering. *Communications in Pure and Applied Mathematics* (2012).

[8] Mohamed, A.-r., Dahl, G. E., and Hinton, G. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on 20*, 1 (2012), 14–22.

[9] Tzanetakis, G., and Cook, P. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on 10*, 5 (2002), 293–302.

[10] Waldspurger, I., d'Aspremont, A., and Mallat, S. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, 1–35.