

# Binaural Scene Classification with Time-Frequency Scattering and Deep Convolutional Networks

Vincent Lostanlen  
École normale supérieure  
45 rue d'Ulm, 75005 Paris, France

**Abstract**—The abstract goes here.

## I. INTRODUCTION

Classification of acoustic scenes is only made possible by integrating signal information over a long temporal context. Whereas a few seconds are often sufficient to recognize a speaker, a musical instrument, or a genre, it may require up to 30 seconds to disambiguate closely related acoustic scenes.

## II. TIME-FREQUENCY SCATTERING

Let  $\psi[t]$  an analytic band-pass filter of dimensionless frequency 1 and bandwidth  $1/Q$ . A filter bank of wavelets is built by dilating  $\psi$  according to a geometric sequence of scales  $2^{-k_1/Q}$ , where the log-frequency index  $k_1$  takes integer values. We denote by  $\psi_{k_1}[t]$  the resulting wavelets. In all subsequent experiments,  $\psi$  is designed as a Gammatone wavelet of quality factor  $Q = 4$ , so as to approximate the properties of the human cochlea. The wavelet transform of an audio signal  $x[t]$  is obtained by convolution with all wavelets:  $y_1[t, k_1] = (x * \psi_{k_1})[t]$ . Applying pointwise complex modulus to  $y_1$  yields the wavelet scalogram  $x_1[t, k_1] = |y_1[t, k_1]|$ , also called constant  $Q$  transform (CQT), indexed by time  $t$  and log-frequency  $k_1$ .

Time-frequency scattering consists in convolving  $x_2[t, k_1]$  with a family of two-dimensional wavelets  $\Psi_{k_2}[t, k_1]$ , where the index  $k_2$  encapsulate two values, i.e. a temporal scale  $\alpha$  and a log-frequential scale  $\beta$ . The temporal scale  $\alpha$  takes 5 values between 23 ms and 370 ms according to a geometric sequence. The log-frequential scale  $\beta$  takes 5 values between  $1/4^{\text{th}}$  of an octave and 4 octaves, as well as its 5 "mirror" frequencies. In addition, the edge case  $\beta = \infty$  corresponds to a log-frequential moving average along 4 octaves according to all 5 temporal scales  $\alpha$ . Finally, the edge case  $(\alpha, \beta) = (\infty, 0)$  corresponds to a temporal moving average  $\phi[t]$  along 370 ms without any transformation of the log-frequency axis. In sum, there are  $5 \times 5 \times 2 + 5 + 1 = 56$  time-frequency scattering features for every time  $t$  and log-frequency  $k_1$ .

The time-frequency scattering coefficients are thus defined as

$$y_2[t, k_1, k_2] = (x_1 \overset{t, k_1}{*} \Psi_{k_2})[t, k_1] = \sum_{\tau, \kappa_1} x_1[t - \tau, k_1 - \kappa_1] \Psi_{k_2}[\tau, \kappa_1]. \quad (1)$$

For some index  $k_2$  involving a temporal scale  $\alpha$ , applying pointwise complex modulus to  $y_2[t, k_1, k_2]$  provides translation-invariant coefficients as long as the amount of translation does not exceed  $\alpha$ .

$$x_2[t, k_1, k_2] = |y_2[t, k_1, k_2]| \overset{t}{*} \phi[t] \quad (2)$$

## III. DEEP CONVOLUTIONAL NETWORKS

Each layer in a convolutional network typically consists in the composition of three operations: two-dimensional convolutions, application of a pointwise nonlinearity, and local pooling.

$$y_3[t, k_1, k_3] = \sum_{k_2} b_3[k_2, k_3] + W_3[t, k_1, k_2, k_3] \overset{t, k_1}{*} x_2[t, k_1, k_2]. \quad (3)$$

We apply the rectified linear unit (ReLU) nonlinearity, with a rectifying slope of  $\alpha = 0.3$  for negative inputs.

$$y_3^+[t, k_1, k_3] = \begin{cases} \alpha y_3[t, k_1, k_3] & \text{if } y_3[t, k_1, k_3] < 0 \\ y_3[t, k_1, k_3] & \text{if } y_3[t, k_1, k_3] \geq 0 \end{cases} \quad (4)$$

At the pooling step, we retain the maximal activation among neighboring units in the time-frequency domain  $(t, k_1)$  over non-overlapping rectangles of width  $\Delta t$  and height  $\Delta k_1$ .

$$x_3[t, k_1, k_3] = \max_{\substack{0 \leq \tau < \Delta t \\ 0 \leq \kappa_1 < \Delta k_1}} \left\{ y_3^+[t - \tau, k_1 - \kappa_1, k_3] \right\} \quad (5)$$

The hidden units in  $x_3$  are in turn fed to a second layer of convolutions, ReLU, and pooling.

$$x[t] = r \times x^L[t] + (1 - r) \times x^R[t], \quad (6)$$

where  $r$  is drawn uniformly at random in the interval  $[0, 1]$ .

## IV. CONCLUSION

The conclusion goes here.