NOTE Communicated by David L. Donoho

A Mathematical Motivation for Complex-Valued Convolutional Networks

Mark Tygert
tygert@fb.com
Joan Bruna
joan.bruna@berkeley.edu
Soumith Chintala
soumith@fb.com
Yann LeCun
yann@fb.com
Serkan Piantino
spiantino@fb.com
Arthur Szlam
aszlam@fb.com

A complex-valued convolutional network (convnet) implements the repeated application of the following composition of three operations, recursively applying the composition to an input vector of nonnegative real numbers: (1) convolution with complex-valued vectors, followed by (2) taking the absolute value of every entry of the resulting vectors, followed by (3) local averaging. For processing real-valued random vectors, complex-valued convnets can be viewed as data-driven multiscale windowed power spectra, data-driven multiscale windowed absolute spectra, data-driven multiwavelet absolute values, or (in their most general configuration) data-driven nonlinear multiwavelet packets. Indeed, complex-valued convnets can calculate multiscale windowed spectra when the convnet filters are windowed complex-valued exponentials. Standard real-valued convnets, using rectified linear units (ReLUs), sigmoidal (e.g., logistic or tanh) nonlinearities, or max pooling, for example, do not obviously exhibit the same exact correspondence with data-driven wavelets (whereas for complex-valued convnets, the correspondence is much more than just a vague analogy). Courtesy of the exact correspondence, the remarkably rich and rigorous body of mathematical analysis for wavelets applies directly to (complex-valued) convnets.

1 Introduction

Convolutional networks (convnets) have become increasingly important to artificial intelligence in recent years, as reviewed by LeCun, Bengio, and Hinton (2015). This note presents a theoretical argument for complex-valued convnets and their remarkable performance. Complex-valued convnets

Neural Computation **28, 815–825** (2016)
doi:10.1162/NECO_a_00824

turn out to calculate "data-driven multiscale windowed spectra" characterizing certain stochastic processes common in the modeling of time series (such as audio) and natural images (including patterns and textures). We motivate the construction of such multiscale spectra using "local averages of multiwavelet absolute values" or, more generally, "nonlinear multiwavelet packets."

A textbook treatment of all concepts and terms we use in this note is given by Mallat (2008). Further information is available in the original work of Daubechies (1992), Meyer (1993), Coifman, Meyer, Quake, and Wickerhauser (1994), Coifman and Donoho (1995), Simoncelli and Freeman (1995), Meyer and Coifman (1997), LeCun, Bottou, Bengio, and Haffner (1998), Donoho, Mallat, von Sachs, and Samuelides (2003), Srivastava, Lee, Simoncelli, and Zhu (2003), Rabiner and Schafer (2007), and Mallat (2008), for example. The work of Haensch and Hellwich (2010), Mallat (2010), Poggio, Mutch, Leibo, Rosasco, and Tacchetti (2012), Bruna and Mallat (2013), Bruna, Mallat, Bacry, and Muzy (2015), and Chintala et al. (2015) also develops complex-valued convnets, providing copious applications and numerical experiments. A related, more sophisticated connection (to renormalization group theory) is given by Mehta and Schwab (2014). Our exposition relies on nothing but the basic signal processing treated by Mallat (2008). Using the connections discussed below, the rich, rigorous mathematical analysis surveyed by Daubechies (1992), Meyer (1993), Mallat (2008), and others applies directly to complex-valued convnets.

Citing such connections, the anonymous reviews of this note suggested viewing complex-valued convnets as a kind of baseline architecture for much of the deep learning reviewed by LeCun et al. (2015). Section 6 presents numerical analyses corroborating this viewpoint. Having such a theoretical basis for deep learning could help in paring down the combinatorial explosion of possibilities for future developments, while probably illuminating further possibilities as well.

The rest of this note proceeds as follows. Section 2 reviews stationary stochastic processes and their spectra. Section 3 reviews locally stationary stochastic processes and the connection of their spectra to stages in a complex-valued convnet. Section 4 introduces multiscale (multiple stages in a convnet). Section 5 describes the fitting (also known as learning or training) that the connection to convnets facilitates. Section 6 briefly compares on a common benchmark the accuracies for the complex-valued convnets of Chintala et al. (2015) to those for the scattering transforms of Mallat (2010) and for the standard real-valued convnets of Krizhevsky, Sutskever, and Hinton (2012). Section 7 generalizes and summarizes the note.

2 Stationary Stochastic Processes

For simplicity, we first limit consideration to the special case of a doubly infinite sequence of nonnegative random variables X_k, where k ranges over the integers. This input data will be the result of convolving an unmeasured independent and identically distributed (i.i.d.) sequence Z_k, where k ranges over the integers, with an unknown sequence of real numbers f_k, where k ranges over the integers (this latter sequence is known as a "filter," whereas the i.i.d. sequence is known as "white noise"):

$$X_j = \sum_{k=-\infty}^{\infty} f_{j-k} Z_k \tag{2.1}$$

for any integer j. Such a sequence X_k, with k ranging over the integers, is a (strictly) "stationary stochastic process." The term *strictly stationary* refers to the fact that lagging or shifting the process preserves the probability distribution of the process: for any integer l, the shift $Y_k = X_{k-l}$, where k ranges over the integers, satisfies

$$Y_j = \sum_{k=-\infty}^{\infty} f_{j-k} Z'_k \tag{2.2}$$

for any integer j, where $Z'_k = Z_{k-l}$; the sequence Z'_k, with k ranging over the integers, is i.i.d. with the same distribution as Z_k, where k ranges over the integers.

The associated "absolute spectrum" is

$$\tilde{X}(\omega) = \lim_{n\to\infty} \mathbf{E} \left| \frac{1}{\sqrt{2n+1}} \sum_{k=-n}^{n} e^{-ik\omega} X_k \right| \tag{2.3}$$

for any real number ω (usually we consider not just any, but instead restrict consideration to a sequence running from 0 to about 2π). Note that lagging or shifting the process changes neither the probability distribution of the process (since the process is stationary) nor the absolute spectrum: for any integer l, the shift $Y_k = X_{k-l}$ yields $\tilde{Y}(\omega) = \tilde{X}(\omega)$ for any real number ω, due to the absolute value in equation 2.3.

Similarly, the associated "power spectrum" is

$$\tilde{\tilde{X}}(\omega) = \lim_{n\to\infty} \mathbf{E} \left(\left| \frac{1}{\sqrt{2n+1}} \sum_{k=-n}^{n} e^{-ik\omega} X_k \right|^2 \right) \tag{2.4}$$

for any real number ω. There is an extra squaring under the expectation in equation 2.4 compared to equation 2.3. Again, lagging or shifting the process changes neither the probability distribution of the process nor the power spectrum. For any integer l, the shift $Y_k = X_{k-l}$ yields $\tilde{Y}(\omega) = \tilde{X}(\omega)$ for any real number ω due to the absolute value in equation 2.4. The remainder of the note focuses on the absolute spectrum; most of the discussion applies to the power spectrum, too.

Remark 1. The absolute spectrum can be more robust than the power spectrum, in the same sense that the mean absolute deviation can be more robust than the variance or standard deviation. The power spectrum is more fundamental in a certain sense, yet the absolute spectrum may be preferable for applications to machine learning. We conjecture that both can work about the same. We focus on the absolute spectrum to simplify the exposition.

3 Locally Stationary Stochastic Processes

In practice, the input data are seldom strictly stationary, but usually only locally stationary, that is, equation 2.1 becomes

$$X_j = \sum_{k=-\infty}^{\infty} f_{j-k}^{(j)} Z_k \tag{3.1}$$

for any integer j, where $f_k^{(j)}$ changes much more slowly when changing j than when changing k. To accommodate such data, we introduce windowed spectra; for any even nonnegative-valued sequence g_k, with k ranging through the integers (this sequence could be samples of a gaussian or any other window suitable for Gabor analysis; the data will determine g during training), we consider

$$\tilde{X}_l(\omega) = \frac{1}{2n+1} \sum_{j=-n+l}^{n+l} \left| \frac{1}{\sqrt{2n+1}} \sum_{k=-\infty}^{\infty} e^{-ik\omega} g_{k-j} X_k \right| \tag{3.2}$$

for any integer l, with some positive integer n. The extra summation in equation 3.2 averages away noise and is a kind of approximation to the expected value in equation 2.3. Usually g_k is fairly close to 1 for $k = -n$, $-n+1, \ldots, n-1, n$, and g_k is fairly close to 0 for $|k| > n$, making a reasonably smooth transition between 0 and 1. The most important difference between equation 2.3 and equation 3.2 is the absence of a limit in the latter (hence the terminology, "local" spectrum).

Due to the absolute value, equation 3.2 is equivalent to

$$\tilde{X}_l(\omega) = \frac{1}{2n+1} \sum_{j=-n+l}^{n+l} \left| \frac{1}{\sqrt{2n+1}} \sum_{k=-\infty}^{\infty} g_{j-k}(\omega)\, X_k \right| \tag{3.3}$$

for any *even* nonnegative-valued sequence g_k, with k ranging through the integers, where

$$g_k(\omega) = e^{ik\omega} g_k \tag{3.4}$$

for any integer k (“even” means that $g_{-k} = g_k$ for every integer k). Note that the right-hand side of equation 3.3 is just a convolution followed by the absolute value followed by local averaging. This will facilitate fitting (learning) using data, enabling a data-driven approach, in section 5.

4 Multiscale

In most cases, the ideal choices of n and width of the window in equation 3.3, that is, the ideal number of indices for which g_k is substantially nonzero, are far from obvious. Often, in fact, multiple widths are relevant (say, wider for lower-frequency variations than for higher frequency). Not knowing the ideal a priori, we use multiple windows on multiple scales. An especially efficient multiscale implementation processes the results of the lowest-frequency channels recursively. For the lowest frequency, $\omega = 0$, and when X_k is nonnegative for every integer k (e.g., the input X_k could be the $\tilde{X}_k$ arising from previous processing), equation 3.3 simplifies to

$$\tilde{X}_l(0) = \frac{1}{\sqrt{2n+1}} \sum_{k=-\infty}^{\infty} h_{l-k} X_k \tag{4.1}$$

for any integer l, where

$$h_l = \frac{1}{2n+1} \sum_{j=-n+l}^{n+l} g_j \tag{4.2}$$

for any integer l, and again g_j, with j ranging through the integers, is an even sequence of nonnegative real numbers (“even” means that $g_{-j} = g_j$ for every integer j). The result of equation 4.1 is simply a convolution with the input sequence, and further convolutions—say, via recursive processing of the form in equation 3.3—can undo this convolution and set the effective window however desired in later stages. The deconvolution and subsequent convolution with the windowed exponential of a later stage is numerically stable if the later window is wider than the preceding. In particular, recursively processing the zero-frequency channels in this way can implement a “wavelet transform” (if each recursive stage considers only two values for ω, one zero and one nonzero—see Figure 1) or a “multiwavelet transform” (if each recursive stage considers multiple values for ω, with one of

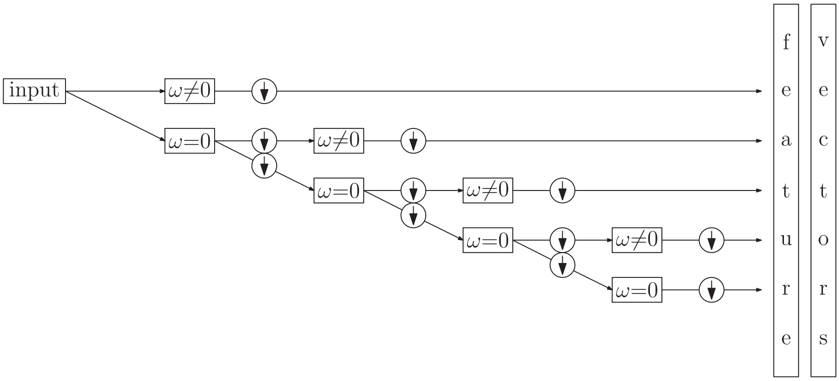

Figure 1: A flowchart for the wavelet transform of an input vector. Each box $\omega = 0$ corresponds to equation 3.3 with $\omega = 0$ or (equivalently) to equation 4.1. Each box $\omega \neq 0$ corresponds to equation 3.3—convolution followed by taking the absolute value of every entry followed by local averaging. Each circle $\downarrow$ corresponds to subsampling (say, retaining only every other entry).

the values being zero—see Figure 2). For multidimensional signals, multiwavelets detect local directionality beyond what wavelets provide. If we recursively process the higher-frequency channels too, then we obtain a "nonlinear wavelet packet transform" or a "nonlinear multiwavelet packet transform," a kind of nonlinear iterated filter bank (see Figure 3). Linearly recombining the different frequency channels may help realize local rotation invariance and other potentially desirable properties (indeed, Mallat, 2010, did this for rotations and other transformations), including generating harmonics when processing audio signals. The transforms just discussed are undecimated, but interleaving appropriate decimation or subsampling applied to the sequences yields the usual decimated transforms.

Remark 2. In practice, decimation or subsampling is important to avoid overfitting in the data-driven approach discussed below, by limiting the number of degrees of freedom appropriately. Even when the signal is not a strictly stationary stochastic process, the averaging in equation 3.3 (the left-most summation) performs the cycle spinning of Coifman and Donoho (1995) to avoid artifacts that would otherwise arise due to windows' partitioning after subsampling. The averaging reduces the variance; wider averaging would further reduce the variance.

Remark 3. Sequences that are finite rather than doubly infinite provide only enough information for estimating a smoothed version of the spectrum. Alternatively, a finite amount of data provides information for estimating multiscale windowed spectra yielding time-frequency (or space-Fourier) resolution similar to the multiresolution analysis of wavelets.

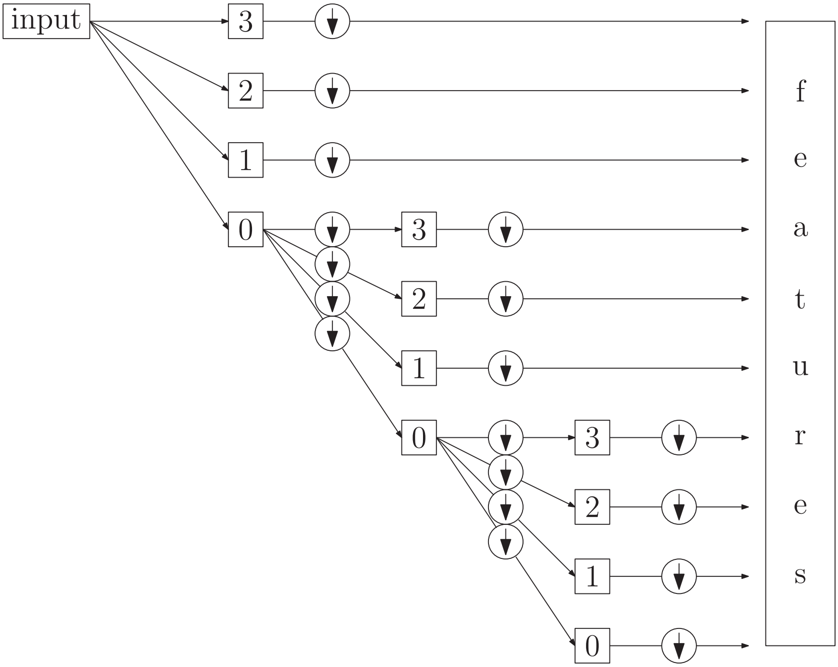

Figure 2: A flowchart for the multiwavelet transform of an input vector. Each box 0 corresponds to equation 3.3 with $\omega = 0$ or (equivalent) to equation 4.1. Each box 1, 2, or 3 corresponds to equation 3.3 for different convolutional filters, but always with convolution followed by taking the absolute value of every entry followed by local averaging. Each circle ↓ corresponds to subsampling (say, retaining only every fourth entry).

Remark 4. SIFT (scale-invariant feature transform), HOG (histograms of oriented gradients), and SURF (speeded-up robust features) of Lowe (1999, 2004), Dalal and Triggs (2005), Bay, Ess, Tuytelaars, and Gool (2008), and others are more analogous to the multiwavelet architecture of Figure 2 than to the more general wavelet-packet architecture of Figure 3.

5 Fitting

The "multiwavelet transform" constitutes a desirable baseline model. We can easily adapt to the data the choices of windows and indeed the whole recursive structure of the processing (whether restricting the recursion to the zero-frequency channels or also allowing the recursive processing of higher-frequency channels). Viewing the convolutional filters in equation 3.3 that serve as windowed exponentials as parameters, the desirable baseline is

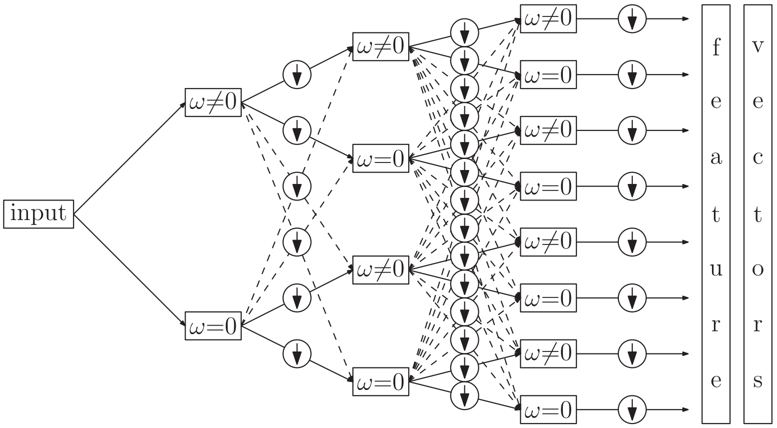

Figure 3: A flowchart for the nonlinear wavelet packet transform of an input vector. Each box $\omega = 0$ corresponds to equation 3.3 with $\omega = 0$ or (equivalently) to equation 4.1. Each box $\omega \neq 0$ corresponds to equation 3.3—convolution followed by taking the absolute value of every entry followed by local averaging. Each circle ↓ corresponds to subsampling (say, retaining only every other entry). The dashed arrows can involve downweighting the associated summands (and the convolutional filter can be different for every arrow). Figure 1 is essentially a special case of this figure for which some of the convolutional filters simply deconvolve the preceding local averaging (omitting some of the subsampling).

just one member of a parametric family of models. This parametric family is known as a "complex-valued convolutional network." We can fit (i.e., learn or train) the parameters to the data using optimization procedures such as stochastic gradient descent in conjunction with "backpropagation" (backpropagation is the chain rule of Calculus applied to calculate gradients of our recursively composed operations). For supervised learning, we optimize according to a specified objective, usually using the multiscale spectra as inputs to a scheme for classification or regression, as detailed by LeCun et al. (1998), for example.

Remark 5. In consonance with the "best-basis" approach of Coifman et al. (1994) and Saito and Coifman (1995), a potentially more efficient possibility is to restrict the convolutional filters in equation 3.3 to be windowed exponentials that are designed completely a priori, aside from one overall scaling factor per filter, fitting only the scaling factors. How best to effect this approach is an open question.

6 Numerical Experiments

This section reports the classification accuracies for the complex-valued convnets of Chintala et al. (2015), the standard real-valued convnets of Krizhevsky et al. (2012), and the scattering transforms of Oyallon and Mallat (2015), on a benchmark data set, CIFAR-10, from Krizhevsky (2009) (CIFAR-10 contains 50,000 images in its training set and 10,000 images in its testing set; each image falls into one of 10 classes, is full color, and consists of a 32×32 grid of pixels). According to Table 4 of Oyallon and Mallat (2015), the scattering transforms attain an error rate of 18% on the test set after training their classifiers on the training set. According to section 3.3 of Krizhevsky et al. (2012), a standard real-valued convnet attains an error rate of 13% on the test set without the local response normalization of that section 3.3 and attains 11% with the local response normalization. The complex-valued convnets detailed in Chintala et al. (2015) attain an error rate of 12% on the test set, at least when using a larger net and training with enough iterations for the test error to settle down and converge (for complex-valued convnets, accuracy seems to improve as the net becomes larger; for the error rate of 12%, a net eight times the size of that reported in Table 1 of Chintala et al., 2015, was sufficient, using the same kernel sizes and other parameter settings as for Table 1). Augmenting the training images with their mirror images improved convergence to the reported accuracies. All in all, the extensively trained real- and complex-valued convnets yielded similar error rates, which are about one-third less than those that scattering transforms attained. Of course, the fitting/learning/training involved for classification with the scattering transforms is much less extensive.

7 Conclusion

While the above concerns X_k, where k ranges over the integers, extending the above to analyze $X_{j,k}$, where j and k range over the integers, is straightforward; the latter could be a "locally homogeneous random field." Also, the infinite range of the integers is far from essential; implementations on computers obviously use only finite sequences. Moreover, the above construction is appropriate for processing any locally stationary stochastic process, not just filtered white noise. For instance, the construction can enable a multiresolution analysis of regularity (or smoothness) that easily distinguishes between low-pass filtered i.i.d. gaussian noise and a pulse train or sinusoid with a random phase offset (e.g., $X_k = 1 + \sin(\pi(k + J)/1000)$ for any integer k, where J is an integer drawn uniformly at random from 1, 2, . . ., 2000). More generally, the construction should enable discriminating between many interesting classes of stochastic processes, commensurate with the ability of multiwavelet-based multiresolution analysis to measure regularity, intermittency, distributional characteristics (say, gaussian versus Poisson), and so on. Any globally stationary stochastic process, with or without intermittent fluctuations, can be modeled as above as a locally stationary stochastic process (of course, Bruna et al., 2015, treat the former directly, to great advantage in the analysis of homogeneous turbulence and other phenomena from statistical physics). Every model in the parametric family constituting the complex-valued convnet calculates relevant features, windowed spectra of the form in equations 3.2 and 3.3. The absolute values in equations 3.2 and 3.3 are the key nonlinearity, a reflection of the local stationarity—the local translation invariance—of the process and its relevant features.

Acknowledgments

We would like to thank Keith Adams, Lubomir Bourdev, Rob Fergus, Armand Joulin, Manohar Paluri, Christian Puhrsch, Marc'Aurelio Ranzato, Ben Recht, Rachel Ward, and the editor and reviewers.

References

Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision Image Understanding*, *110*(3), 346–359.

Bruna, J., & Mallat, S. (2013). Invariant scattering convolutional networks. *IEEE Trans. Pattern Analysis Machine Intel.*, *35*(8), 1872–1886.

Bruna, J., Mallat, S., Bacry, E., & Muzy, J.-F. (2015). Intermittent process analysis with scattering moments. *Ann. Statist.*, *43*(1), 323–351.

Chintala, S., Ranzato, M., Szlam, A., Tian, Y., Tygert, M., & Zaremba, W. (2015). *Scale-invariant learning and convolutional networks* (Tech. Rep.). 1506.08230, arXiv.

Coifman, R. R., & Donoho, D. (1995). Translation-invariant denoising. In A. Antoniadis & G. Oppenheim (Eds.), *Wavelets and statistics* (pp. 125–150). New York: Springer.

Coifman, R. R., Meyer, Y., Quake, S., & Wickerhauser, M. V. (1994). Signal processing and compression with wavelet packets. In J. S. Byrnes, J. L. Byrnes, K. A. Hargreaves, & K. Berry (Eds.), *Wavelets and their applications* (pp. 363–379). New York: Springer.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conf. Computer Vision and Pattern Recognition 2005* (vol. 1, pp. 886–893). Piscataway, NJ: IEEE.

Daubechies, I. (1992). *Ten lectures on wavelets*. Philadelphia: SIAM.

Donoho, D., Mallat, S., von Sachs, R., & Samuelides, Y. (2003). Locally stationary covariance and signal estimation with macrotiles. *IEEE Trans. Signal Processing*, *51*(3), 614–627.

Haensch, R., & Hellwich, O. (2010). Complex-valued convolutional neural networks for object detection in PolSAR data. In *Proceedings of the 8th European Conf. EUSAR* (pp. 1–4). Piscataway, NJ: IEEE.

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images.* Master's thesis, University of Toronto.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, 25 (pp. 1097–1105). Red Hook, NY: Curran.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, *86*(11), 2278–2324.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE Internat. Conf. Computer Vision* (vol. 2, pp. 1150–1157). Piscataway, N.J: IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Internat. J. Computer Vision*, *60*(2), 91–110.

Mallat, S. (2008). *A wavelet tour of signal processing: The sparse way* (3rd ed.). Orlando, FL: Academic Press.

Mallat, S. (2010). Recursive interferometric representations. In *Proc. of the EUSIPCO Conf. 2010* (pp. 716–720). Piscataway, NJ: IEEE.

Mehta, P., & Schwab, D. J. (2014). *An exact mapping between the variational renormalization group and deep learning*. (Tech. Rep.). 1410.3831, arXiv.

Meyer, Y. (1993). *Wavelets and operators*. Cambridge: Cambridge University Press.

Meyer, Y., & Coifman, R. R. (1997). *Wavelets: Calderón-Zygmund and multilinear operators*. Cambridge: Cambridge University Press.

Oyallon, E., & Mallat, S. (2015). Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Computer Society Conf. Computer Vision and Pattern Recognition 2015* (vol. 1, pp. 2865–2873). Piscataway, NJ: IEEE.

Poggio, T., Mutch, J., Leibo, J., Rosasco, L., & Tacchetti, A. (2012). *The computational magic of the ventral stream: Sketch of a theory (and why some deep architectures work)* (Tech. Rep. MIT-CSAIL-TR-2012-035), Cambridge, MA: MIT CSAIL.

Rabiner, L. R., & Schafer, R. W. (2007). *Introduction to digital speech processing.* Hanover, MA: Now Publishers.

Saito, N.& Coifman, R. R. (1995). Local discriminant bases and their applications. *J. Math. Imaging Vision*, *5*(4), 337–358.

Simoncelli, E. P., & Freeman, W. T. (1995). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proceedings of the Internat. Conf. Image Processing 1995* (vol. 3, pp. 444–447). Piscataway, NJ: IEEE.

Srivastava, A., Lee, A. B., Simoncelli, E. P., & Zhu, S. (2003). On advances in statistical modeling of natural images. *J. Math. Imaging Vision*, *18*(1), 17–33.

Received October 11, 2015; accepted December 13, 2015.