To appear: Proc. 21st IEEE Int'l Conf on Image Processing, Oct 27-30, 2014, Paris, France.
Published by: IEEE Computer Society.

LEARNING SPARSE FILTER BANK TRANSFORMS WITH CONVOLUTIONAL ICA

Johannes Ballé, Eero P. Simoncelli

Center for Neural Science/Courant Institute of Mathematical Sciences
and Howard Hughes Medical Institute, New York University
{johannes.balle,eero.simoncelli}@nyu.edu

ABSTRACT

Independent Component Analysis (ICA) is a generalization of Principal Component Analysis that optimizes a linear transformation to whiten and sparsify a family of source signals. The computational costs of ICA grow rapidly with dimensionality, and application to high-dimensional data is generally achieved by restricting to small windows, violating the translation-invariant nature of many real-world signals, and producing blocking artifacts in applications. Here, we reformulate the ICA problem for transformations computed through convolution with a bank of filters, and develop a generalization of the fastICA algorithm for optimizing the filters over a set of example signals. This results in a substantial reduction of computational complexity and memory requirements. When applied to a database of photographic images, the method yields bandpass oriented filters, whose responses are sparser than those of orthogonal wavelets or block DCT, and slightly more heavy-tailed than those of block ICA, despite fewer model parameters.

Index Terms— independent component analysis, stationarity, sparsity, convolutional filters, fastICA, filter bank

1. INTRODUCTION

Independent Component Analysis (ICA) refers to a family of algorithms devised to solve the blind source separation problem. One observes a set of vectors assumed to arise from a linear transformation of a set of independent sources with non-Gaussian marginal densities, and seeks to estimate the linear mixing matrix from these observations, which may then be used to estimate the source vector for subsequent observations. A variety of solutions to this problem may be derived under different formulations: as a maximum likelihood estimation [1], as a method to maximize information transmission in a noiseless neural network [2], as a minimization of mutual information between the "unmixed" outputs [3], or as a higher-order decorrelation based on the cumulant tensor [4]. Algorithms which are tractable for high dimensional data can generally be divided into those based on descent of the natural gradient [5], and those that are variants of the fastICA algorithm [3], which is popular for its computational efficiency.

In practice, ICA has been used to develop optimal decompositions for relatively small, fixed-size, data vectors (e.g., the image intensities within a small rectangular window). This enables efficient computation, both because the computations per iteration scale with vector size, and because the training set size required to avoid overfitting grows super-linearly with dimensionality. But the statistical properties of many signals of interest are translation-invariant, and windowing the training data introduces artificial boundaries. Consequently, signal values near block boundaries are represented differently than those near the block center. When used in applications such as compression or restoration, lack of translation invariance results in blocking artifacts, a problem well-known to anyone who has viewed a severely-compressed JPEG image.

Here, we reformulate the ICA problem for transforms computed through convolution/subsampling with a bank of filters, a form that has been ubiquitous in signal and image processing for the past 30 years (e.g., [6], [7]). The convolutional structure allows for substantially fewer model parameters, without imposing an arbitrary subdivision of the signal into blocks. Similar advantages have been exploited in the development of convolutional artificial neural networks used for image classification and recognition [8]. The algorithm converges quickly to a stable solution, and is significantly more efficient than the fastICA algorithm. When applied to photographic images, it yields bandpass oriented filters that tile the frequency plane, but that are qualitatively different than those of traditional filter banks, or those obtained through blocked PCA or ICA.

2. CONVOLUTIONAL ICA

A common formulation of the ICA problem is in terms of information-theoretic quantities [2], [3]. We seek an "unmixing" matrix $\hat{\boldsymbol{W}}$, transforming an N-sample observation vector $\boldsymbol{x}$ so as to minimize the multi-information between its elements:

$$\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} I(\boldsymbol{W}\boldsymbol{x}), \tag{1}$$

where

$$I(\boldsymbol{y}) = \sum_i H(y_i) - H(\boldsymbol{y}) \tag{2}$$

$$H(\boldsymbol{y}) = -\,\mathbb{E}_{\boldsymbol{y}} \log p_{\boldsymbol{y}}(\boldsymbol{y})$$

are multi-information and differential entropy, respectively.

Optimization of entropy-related quantities is notoriously difficult, and ICA methods typically proceed by introducing restrictions and simplifications. In particular, the fastICA method [3] relies on two such simplifications. First, the solution is constrained to the form $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{C}_{\boldsymbol{x}}^{-\frac{1}{2}}$, where $\boldsymbol{U}$ is an orthogonal matrix, $\boldsymbol{C}_{\boldsymbol{x}}$ is the covariance of the data, and $\boldsymbol{C}_{\boldsymbol{x}}^{-\frac{1}{2}}$ is a whitening matrix.[1] As a result, the optimization problem of (1) becomes:

$$\hat{\boldsymbol{U}} = \arg\min_{\boldsymbol{U}} I(\boldsymbol{U}\boldsymbol{z}), \tag{3}$$

where $\boldsymbol{z} = \boldsymbol{C}_{\boldsymbol{x}}^{-\frac{1}{2}}\boldsymbol{x}$ is the prewhitened data. Given the orthogonality of $\boldsymbol{U}$, the second term of (2) becomes:

$$H(\boldsymbol{U}\boldsymbol{C}_{\boldsymbol{x}}^{-\frac{1}{2}}\boldsymbol{x}) = \underbrace{\log|\boldsymbol{U}|}_{=0} - \frac{1}{2}\log|\boldsymbol{C}_{\boldsymbol{x}}| + H(\boldsymbol{x}), \tag{4}$$

which is independent of $\boldsymbol{U}$. Thus, the optimization problem reduces to minimizing the sum of marginal entropies (first term in (2)). A second simplification facilitates this optimization: the marginal entropies are approximated with a more tractable *contrast function* such as $\tilde{H}(y_k) = \log\cosh(y_k)$, which seeks heavy-tailed (i.e., sparse) components in the data.

To achieve a transform that respects the assumed stationarity of the input signal, we constrain the matrix $\boldsymbol{W}$ to a form corresponding to convolution with a bank of filters. First, we use a whitening matrix implemented as a convolution with a zero-phase filter whose Fourier spectrum is the inverse of the signal spectrum (referred to as *Zero-phase Components Analysis* [2]). Furthermore, we assume $\boldsymbol{U}$ is implemented as a filter bank, and thus is composed of K square D-dimensional Toeplitz matrices $\boldsymbol{B}_k$:

$$\begin{bmatrix}\boldsymbol{y}_0^{\mathsf{T}} & \cdots & \boldsymbol{y}_{K-1}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}} = \begin{bmatrix}\boldsymbol{B}_0^{\mathsf{T}} & \cdots & \boldsymbol{B}_{K-1}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}\boldsymbol{z}, \tag{5}$$

where D is the number of stationary dimensions (i.e., 2 for images) and the output vector $\boldsymbol{y}$ contains the concatenated subbands, $y_k(\boldsymbol{n})$, each resulting from convolution with an associated filter $b_k(\boldsymbol{n})$:

$$y_k(\boldsymbol{n}) = (b_k * z)(\boldsymbol{n}). \tag{6}$$

For $K > 1$, $\boldsymbol{U}$ will be overcomplete (more rows than columns), which is incompatible with the orthogonality constraint. Here, we consider two ways of resolving the conflict: relaxing the constraint, and reducing dimensionality by regular subsampling after the filtering. The derivation of the latter has to be omitted due to spatial constraints, but we summarize the algorithm at the end of the next section. To relax the constraint, a natural choice is to require that $\boldsymbol{U}$ is an *isometry*, meaning that it preserves distances and angles, and thus has singular values all equal to one. If $\boldsymbol{U}$ is overcomplete and isometric, we can always augment it with additional columns $\boldsymbol{V}$ to form an orthogonal matrix such that

$$\boldsymbol{U}\boldsymbol{z} = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{V}\end{bmatrix}\begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{0}\end{bmatrix}. \tag{7}$$

Then,

$$H(\boldsymbol{U}\boldsymbol{z}) = \log 1 - H\left(\begin{bmatrix}\boldsymbol{z}^{\mathsf{T}} & \boldsymbol{0}\end{bmatrix}^{\mathsf{T}}\right).$$

Note that the second term is negative infinity, reflecting the fact that vectors arising from an overcomplete linear transform must be confined to a subspace. Nevertheless, both parts are again independent of $\boldsymbol{U}$; hence, the optimization is separable into univariate subproblems as in fastICA.

The constraint can be written in terms of the filters as

$$\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U} = \sum_k \boldsymbol{B}_k^{\mathsf{T}}\boldsymbol{B}_k = \boldsymbol{1}. \tag{8}$$

For optimization purposes, we set the size of the observations equal to the filter support (for images, a $\sqrt{N}\times\sqrt{N}$-sample window). Compared to ICA, the filters are thus still constrained to a finite support, but the structure of the transform now reflects the fact that the filters must cover the stationary signal dimensions. If N is chosen large enough, we can approximate the Toeplitz blocks as circulant blocks. With this, the constraint can be re-written in the Fourier domain:

$$\forall \boldsymbol{f} : \sum_k |\tilde{b}_k(\boldsymbol{f})|^2 = 1, \tag{9}$$

where $\tilde{b}_k(\boldsymbol{f}) = \mathcal{F}\big(b_k(\boldsymbol{n})\big)$ are the discrete Fourier transforms of the filters.

3. ALGORITHM

To find a solution to (3), we follow the same approach as in the fastICA algorithm [3], alternating between optimizing $H(y_k)$ individually and enforcing the constraints on $\boldsymbol{U}$, until convergence. However, as $\boldsymbol{U}$ is highly redundant, we avoid working with it directly, to obtain a faster algorithm.

First, to enforce isometry, we apply the Fourier-domain constraint (9) by a simple renormalization of the Fourier magnitudes of the filters. To ensure that none of the filters degrade to zero, we scale the filters to unit energy prior to the normalization. Second, to minimize $H(y_i)$ in the objective, we adopt the fixed-point iteration from [3]. Without loss of generality, we can ignore all shifted copies of the filter except one, so that we only need to perform the iteration once for each subband:

$$\boldsymbol{b}_k \leftarrow \boldsymbol{b}_k\,\mathbb{E}_{\boldsymbol{z}}\big(g'(\boldsymbol{b}_k^{\mathsf{T}}\boldsymbol{z})\big) - \mathbb{E}_{\boldsymbol{z}}\big(\boldsymbol{z}g(\boldsymbol{b}_k^{\mathsf{T}}\boldsymbol{z})\big), \tag{10}$$

where $\boldsymbol{b}_k$ is the filter $b_k(\boldsymbol{n})$ vectorized over its support, and g and g' are the first and second derivative of the the contrast function, respectively. Note that the contrast functions used in the literature typically assume that $\|\boldsymbol{b}_k\|_2 = 1$. This is

[1] The multi-information objective can be decomposed into a correlation and a non-Gaussianity term with equal weights [9, eq. 18]. The whitening constraint effectively puts a very large weight on the correlation term.

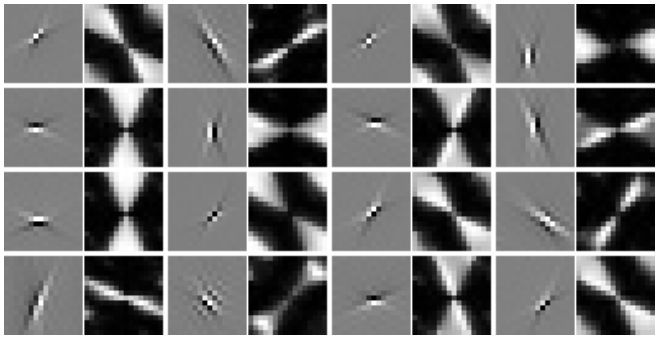

Fig. 1. Complete CICA filter bank, computed for $K = 16$ and subsampling $s = 4$ along with the Fourier magnitudes. The CICA transform consists of a whitening combined with the filters shown here, whereas inversion of the transform requires filtering and unwhitening.

only the case for complete or undercomplete $\boldsymbol{U}$, thus renormalizing the rows before computing this step is an easy way of improving convergence.

The filter coefficients can be translated arbitrarily without affecting the result of the optimization, provided they remain within the support. However, centering the filters within the support helps to maximize the accuracy of the circulant approximation introduced above. This can be done by minimizing their *group delay*

$$d_k(\boldsymbol{n}) = \sum_{\boldsymbol{f}} \left|\tilde{b}_k(\boldsymbol{f})\right| \cdot \left| \frac{\partial}{\partial \boldsymbol{f}} \arg\left(\tilde{b}_k(\boldsymbol{f}) e^{j2\pi \boldsymbol{f}^\mathsf{T} \boldsymbol{n}}\right)\right|, \quad (11)$$

where $\boldsymbol{n}$ is a discrete-valued shift index (which is circular in the filter support). We find that using the principal value of the complex argument, as well as a simple finite difference approximation of the gradient, works well in practice. Because the shift is circular, shifting by too much in one iteration can result in instabilities. Constraining each shift to a maximum of one pixel prevents this and ensures a smooth interaction within the optimization algorithm.

Collecting all the steps, we can summarize each iteration of the convolutional ICA (CICA) algorithm as follows:

1. Fixed point iteration:
   (a) $\forall k : \boldsymbol{b}_k \leftarrow \left(\|\boldsymbol{b}_k\|_2\right)^{-1} \boldsymbol{b}_k$
   (b) $\forall k : \boldsymbol{b}_k \leftarrow \boldsymbol{b}_k \, \mathbb{E}_{\boldsymbol{z}}\left(g'(\boldsymbol{b}_k^\mathsf{T}\boldsymbol{z})\right) - \mathbb{E}_{\boldsymbol{z}}\left(\boldsymbol{z} g(\boldsymbol{b}_k^\mathsf{T}\boldsymbol{z})\right)$
2. Enforce isometry:
   (a) $\forall k : \boldsymbol{b}_k \leftarrow \left(\|\boldsymbol{b}_k\|_2\right)^{-1} \boldsymbol{b}_k$
   (b) $\forall \boldsymbol{f} : \tilde{b}_k(\boldsymbol{f}) \leftarrow \tilde{b}_k(\boldsymbol{f}) \left(\sum_k |\tilde{b}_k(\boldsymbol{f})|^2\right)^{-\frac{1}{2}}$
3. Optimize group delay:
   (a) $\forall k : \hat{\boldsymbol{n}}_k \leftarrow \arg\min_{\boldsymbol{n}} d_k(\boldsymbol{n})$ over $\boldsymbol{n} \in \{-1, 0, 1\}^D$
   (b) $\forall k, \boldsymbol{f} : \tilde{b}_k(\boldsymbol{f}) \leftarrow \tilde{b}_k(\boldsymbol{f}) \exp(j2\pi \boldsymbol{f}^\mathsf{T} \hat{\boldsymbol{n}}_k)$

If an overcomplete transform is undesired, the algorithm can be modified to ensure orthogonality of $\boldsymbol{U}$ by regular subsampling. Then, the following steps need to be replaced:

2. (b) $\forall \boldsymbol{f}$ such that $-\frac{1}{2s} \le f_i < \frac{1}{2s}$, construct a matrix of all aliased frequency bins $\tilde{b}_k(\boldsymbol{f} + \frac{1}{s}\boldsymbol{m})$, enumerating k along the rows and all values of $\boldsymbol{m}$ along the columns. Set all singular values to one.

3. (a) $\forall k : \hat{\boldsymbol{n}}_k \leftarrow \arg\min_{\boldsymbol{n}} d_k(\boldsymbol{n})$ over $\boldsymbol{n} \in \{-s, 0, s\}^D$

where s is the subsampling factor. If we set $s = \sqrt[D]{K}$, the result is a complete filter bank. By choosing arbitrary values of s and K, the overcompleteness of the transform can be adjusted. Analogously, an extension to multiple input channels is possible, such that, for example, RGB images or stereophonic audio can be handled.

4. EXPERIMENTAL RESULTS

We implemented a stochastic version of the algorithm along with an analogous implementation of fastICA, evaluating the expectations in step 1(b) over a new set of random samples in each iteration. All experiments were performed on a natural image database [10].

Figure 1 shows all filters $b_k(\boldsymbol{n})$ and their Fourier spectra for CICA with $K = 16$ and a subsampling factor of $s = 4$ in each direction (i.e., $\boldsymbol{U}$ is orthogonal). Unlike traditional filter banks, the analysis and synthesis filters are not identical up to a transpose, as they require whitening and unwhitening, respectively. Note that, like ICA, the algorithm is not guaranteed to find a global minimum, but we find that results appear qualitatively similar with different initializations. The filters are generally nonuniform in angular bandwidth, and quite broad in their spatial frequency response. Some of the high frequency filters have near-identical Fourier amplitudes, and appear as copies of each other shifted by amounts less than the subsampling factor, suggesting a finer subsampling grid. At values of K roughly greater than 12, the filters begin to inhabit multiple scales, although this is difficult to observe due to their radial broadness. Experiments with larger K and on RGB data are shown at `http://www.cns.nyu.edu/~lcv/cica` along with the full source code in the form of an IPython notebook [11].

It can be misleading to compare the multi-information objective for linear transforms with different numbers of components, especially in the overcomplete case. We avoid the problem by making comparisons only for complete transforms and equal numbers of components. We compared a CICA filter bank with $K = 16$ filters and subsampling of $s = 4$, amounting to a total of 256 components over a support of 16×16 pixels, to an ICA matrix using the same support, number of components, and contrast function. As shown in figure 2, CICA yields overall slightly smaller values of the contrast

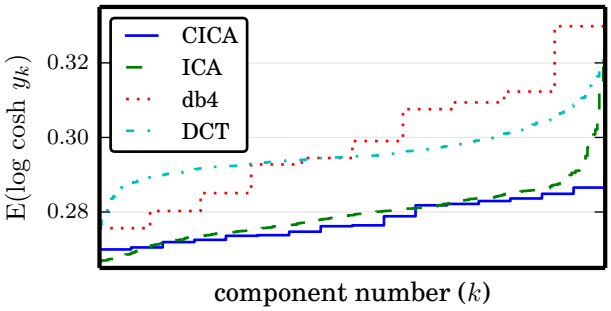

Fig. 2. Comparison of response sparseness of 16×16 DCT, Daubechies-4 wavelet, complete ICA (256 filters) and complete CICA ($K = 16$, $s = 4$).

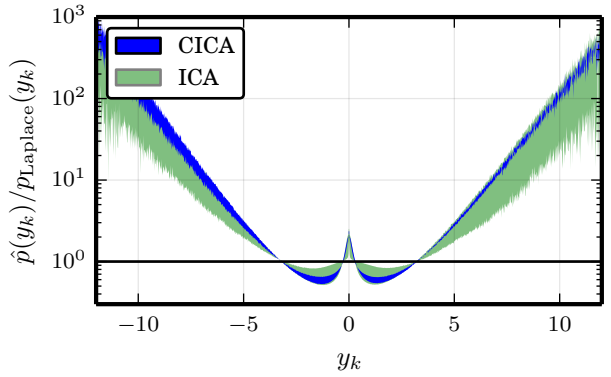

Fig. 3. Histograms of the sparsest and least sparse components of block ICA (top and bottom border of shaded light green area) and CICA (borders of dark blue area) relative to a Laplacian density (black line).

function, despite the fact that the number of trained parameters is $N/K = 16$ times lower than for ICA. This result is consistent with figure 3, in which the histograms of the sparsest and least sparse components of ICA vs. CICA are compared. The sparsest of both methods appear similarly heavy-tailed, whereas the least sparse components of ICA have significantly lighter tails. The latter tend to correspond to components near the block boundary.

Finally, we compared the computational complexity of the two algorithms for the complete case. The reduction of parameters in CICA yields a proportional reduction in memory requirements. As step 1 of the algorithm is adopted unchanged from fastICA, the complexity of this part is reduced by this same factor. The rest of the algorithm is dominated by the enforcement of the isometry constraint. This requires $\mathcal{O}(K)$ FFT operations of size N, each $\mathcal{O}(N \log N)$; and $\mathcal{O}(N/K)$ SVD operations of size $K \times K$, each $\mathcal{O}(K^3)$. The complexity is therefore $\mathcal{O}(KN \log N + NK^2)$. In contrast, block ICA requires one orthogonalization operation of size $N \times N$. Orthogonalization can be achieved either by an eigenvalue decomposition, by a Gram-Schmidt-like deflation scheme, or by an iterative scheme [3]. All three methods can be characterized as $\mathcal{O}(N^3)$. We empirically compared the

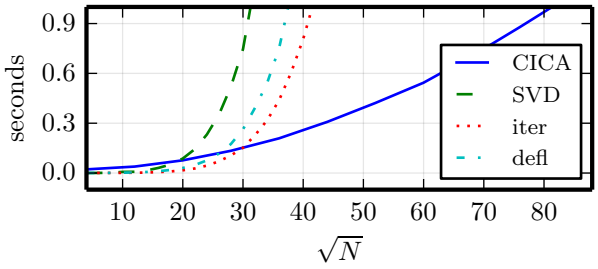

Fig. 4. Empirical comparison of CPU timings for steps 2 and 3 (CICA) vs. three methods of orthogonalization.

CPU time spent in one execution of the latter three methods to the overall CPU time spent executing steps 2 and 3 of the CICA algorithm for $K = 16$ and $s = 4$. As evident from figure 4, CICA is significantly more efficient than fastICA.

5. CONCLUSION

We have developed CICA, a convolutional formulation of ICA, which is tailored for data with stationary statistics, and yields a transform that is structured as a filter bank with adjustable overcompleteness, avoiding the artificial blocking that is typically imposed on ICA solutions. The optimization algorithm is analogous to that of fastICA [3], but is approximately N/K times more efficient (when optimizing a complete filter bank), where N is the size of the data vectors and K is the number of subbands. When applied to a database of natural images, the CICA-optimized filters produce coefficients with heavier-tailed distributions than those of standard orthogonal wavelet or block DCT filters, and slightly better ones than fastICA, despite a reduced parameter space.

Although the use of filter banks is widespread throughout all branches of signal and image processing, they are generally not optimized for source statistics, but instead designed to satisfy a set of desired constraints such as tiling of the frequency domain, orthogonality, critical sampling, minimal aliasing, vanishing moments, or efficient cascaded construction. One exception is the work of Sallee and Olshausen [12], who optimized the sparsity of a multi-scale filter bank subject to the subsampling structure of a pre-chosen multi-scale transform (a steerable pyramid [13]). This constraint, however, resulted in filters that were quite similar to those of the original transform. The CICA formulation is more flexible, and when applied to photographic images, the algorithm yields a set of oriented bandpass filters that tile the Fourier plane differently than those of conventional wavelets or other multi-scale representations. We are currently exploring how these solutions, as well as their usefulness in a number of applications, depend on the choice of constraints (whitening, number of filters, subsampling factor, contrast function). We believe that signal-optimized filter banks have the potential to give substantial improvements across a broad range of image and signal processing problems.

6. REFERENCES

[1] D. T. Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, Jul. 1997. DOI: 10.1109/78.599941.

[2] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995. DOI: 10.1162/neco.1995.7.6.1129.

[3] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, May 1999. DOI: 10.1109/72.761722.

[4] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proceedings F*, vol. 140, no. 6, pp. 362–370, Dec. 1993.

[5] S.-i. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, vol. 8, 1995, pp. 757–763.

[6] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, 1983, ISBN: 0-13-605162-6.

[7] M. Vetterli, "A theory of multirate filter banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 356–372, Mar. 1987. DOI: 10.1109/TASSP.1987.1165137.

[8] Y. LeCun, O. Matan, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, and H. S. Baird, "Handwritten zip code recognition with multilayer networks," in *Proc. of the International Conference on Pattern Recognition*, (Atlantic City), vol. 2, 1990, pp. 35–40.

[9] J.-F. Cardoso, "Dependence, correlation and gaussianity in independent component analysis," *Journal of Machine Learning Research*, vol. 4, pp. 1177–1203, 2003.

[10] G. Tkačik, P. Garrigan, C. Ratliff, G. Milčinski, J. M. Klein, L. H. Seyfarth, P. Sterling, D. H. Brainard, and V. Balasubramanian, "Natural images from the birthplace of the human eye," *PLoS one*, vol. 6, no. 6, Jun. 2011. DOI: 10.1371/journal.pone.0020409.

[11] F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 21–29, May 2007, ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.53.

[12] P. Sallee and B. Olshausen, "Learning sparse multiscale image representations," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15, 2003, pp. 1327–1334.

[13] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: a flexible architecture for multi-scale derivative computation," in *Proc. of IEEE International Conference on Image Processing ICIP '95*, vol. 3, Oct. 1995, pp. 444–447. DOI: 10.1109/ICIP.1995.537667.