Interpretable and robust blind image denoising with bias-free convolutional neural networks

Zahra Kadkhodaie*
Center for Data Science
New York University
zk388@nyu.edu

Sreyas Mohan*
Center for Data Science
New York University
sm7582@nyu.edu

Eero P. Simoncelli
Center for Neural Science, and
Howard Hughes Medical Institute
New York University
eero.simoncelli@nyu.edu

Carlos Fernandez-Granda
Center for Data Science, and
Courant Inst. Mathematical Sciences
New York University
cfgranda@cims.nyu.edu

Abstract

Deep convolutional networks often append additive constant ("bias") terms to their convolution operations, enabling a richer repertoire of functional mappings. Biases are also used to facilitate training, by subtracting mean response over batches of training images (a component of "batch normalization"). Recent state-of-the-art blind denoising methods seem to require these terms for their success. Here, however, we show that bias terms used in most CNNs (additive constants, including those used for batch normalization) interfere with the interpretability of these networks, do not help performance, and in fact prevent generalization of performance to noise levels not including in the training data. In particular, bias-free CNNs (BF-CNNs) are locally linear, and hence amenable to direct analysis with linear-algebraic tools. These analyses provide interpretations of network functionality in terms of projection onto a union of low-dimensional subspaces, connecting the learning-based method to more traditional denoising methodology. Additionally, BF-CNNs generalize robustly, achieving near-state-of-the-art performance at noise levels well beyond the range over which they have been trained.

1 Introduction

Denoising – recovering a signal from measurements corrupted by noise – is a canonical application of statistical estimation that has been studied since the 1950's. Achieving high-quality denoising results requires (at least implicitly) quantifying and exploiting the differences between signals and noise. In the case of natural photographic images, the denoising problem is both an important application, as well as a useful test-bed for our understanding of natural images.

The classical solution to the denoising problem is the Wiener filter (13), which assumes a translation-invariant Gaussian signal model. Under this prior, the Wiener filter is the optimal estimator (in terms of mean squared error). It operates by mapping the noisy image to the frequency domain, shrinking the amplitude of all components, and mapping back to the signal domain. In the case of natural images, the high-frequency components are shrunk more aggressively than the lower-frequency components because they tend to contain less energy in natural images. This is equivalent to convolution with a lowpass filter, implying that each pixel is replaced with a weighted average over a local neighborhood.

In the 1990's, more powerful solutions were developed based on multi-scale ("wavelet") transforms. These transforms map natural images to a domain where they have sparser representations. This makes it possible to perform denoising by applying nonlinear thresholding operations in order to reduce or discard components that are small relative to the noise level (4; 12; 1). From a linear-algebraic perspective, these algorithms operate by projecting the noisy input onto a lower-dimensional subspace that contains plausible signal content. The projection eliminates the orthogonal complement of the subspace, which mostly contains noise. This general methodology laid the foundations for the state-of-the-art models in the 2000's (e.g. (3)), some of which added a data-driven perspective, learning sparsifying transforms (5), or more general nonlinear shrinkage functions directly from natural images (6; 10).

*Equal contribution.

Deep Inverse Workshop, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

In the past decade, purely data-driven models based on convolutional neural networks (8) have come to dominate all previous methods in terms of performance. These models consist of cascades of convolutional filters, and rectifying nonlinearities, which are capable of representing a diverse and powerful set of functions. Training such architectures to minimize mean square error over large databases of noisy natural-image patches achieves current state-of-the-art results (14) (see also (2) for a related approach).

Neural networks have achieved particularly impressive results on the *blind* denoising problem, in which the noise amplitude is unknown (14; 15; 9). Despite their success, We lack intuition about the denoising mechanisms these solutions implement. Network architecture and functional units are often borrowed from the image-recognition literature, and it is unclear which of these aspects contribute positively, or limit, the denoising performance. Many authors claim critical importance of specific aspects of architecture (e.g., skip connections, batch normalization, recurrence), but the benefits of these attributes are difficult to isolate and evaluate in the context of the many other elements of the system.

In this work, we show that bias terms used in most CNNs (additive constants, including those used for batch normalization) interfere with the interpretability of these networks, do not help performance, and in fact prevent generalization of performance to noise levels not including in the training data. In particular, bias-free CNNs (BF-CNNs) are locally linear, and hence amenable to direct analysis with linear-algebraic tools. And BF-CNNs generalize robustly, achieving near-state-of-the-art performance at noise levels well beyond the range over which they have been trained.

2 Analysis and generalization of bias-free neural networks for denoising

We assume a measurement model in which images are corrupted by additive noise: $y = x + n$, where $x \in \mathbb{R}^N$ is the original image, containing N pixels, n is an image of i.i.d. samples of Gaussian noise with variance σ^2, and $y \in \mathbb{R}^N$ is the observed noisy image. The denoising problem consists of finding a function $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$ that provides a good estimate of the original image, x. Commonly, one minimizes the mean squared error : $f(y) = \arg\min_g \mathbb{E}||x - g(y)||^2$, where the expectation is taken over some distribution over images, x, as well as over the distribution of noise realizations. Finally, if the noise standard deviation, σ, is unknown, the expectation should also be taken over a distribution of this variable. This problem is often called *blind denoising* in the literature.

Feedforward neural networks with rectified linear units (ReLUs) are piecewise affine: for a given input signal, the effect of the network on the input is a cascade of linear transformations (convolutional or fully connected layers, each represented by a matrix, (W), additive constants (b), and pointwise multiplication by a binary mask representing the sign of the affine responses (R). Since each stage is affine, the entire cascade implements a single affine transformation. The function computed by a denoising neural network with L layers may be written

$$f(y) = W_L R(W_{L-1}...R(W_1 y + b_1) + ...b_{L-1}) + b_L = A_y y + b_y, \tag{1}$$

where $A_y \in \mathbb{R}^{N \times N}$ is the Jacobian of $f(\cdot)$ evaluated at input y, and $b_y \in \mathbb{R}^N$ represents the net additive bias. The subscripts on A_y and b_y serve as a reminder that the corresponding matrix and vector, respectively, depend on the ReLU activation patterns, which in turn depend on the input vector y.

If we remove all the additive ("bias") terms from every stage of a CNN, the resulting bias-free CNN (BF-CNN) is strictly linear, and its net action may be expressed as

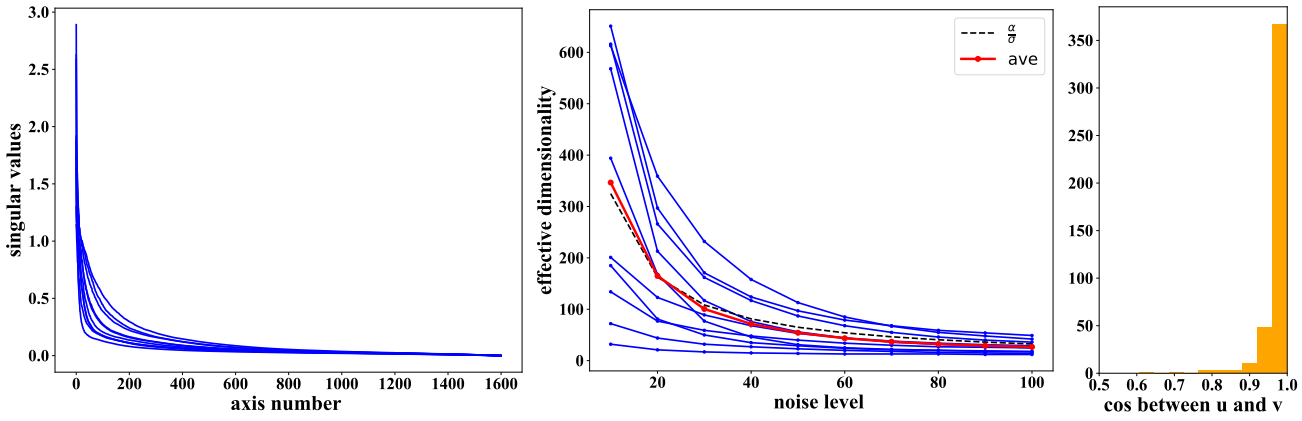
$$f_{\mathrm{BF}}(y) = W_L R(W_{L-1}...R(W_1 y) = A_y y, \tag{2}$$

where A_y is again the Jacobian of $f_{\mathrm{BF}}(\cdot)$ evaluated at y. We analyze this local representation to reveal and visualize the noise-removal mechanisms implemented by BF-CNNs. We illustrate our analysis using a BF-CNN based on the architecture of the Denoising CNN (DnCNN, (14)), although our observations also hold for other architectures (7; 11; 15).

The linear representation of the denoising map given by equation 2 implies that the ith pixel of the output image is computed as an inner product between the ith row of A_y and the input image. The rows of A_y can be interpreted as *adaptive filters* that produce an estimate of the denoised pixel via a weighted average of noisy pixels. Examination of these filters reveals their diversity, and their relationship to the underlying image content: they are adapted to the local features of the noisy image, averaging over homogeneous regions of the image without blurring across edges (Figure 2). We observe that the equivalent filters of all architectures adapt to image structure.

The local linear structure of a BF-CNN allows analysis of its functional capabilities via the singular value decomposition (SVD). For a given input y, we compute the SVD of the Jacobian matrix: $A_y = USV^T$: The output is a linear combination of the left singular vectors, each weighted by the projection of the input onto the corresponding right singular vector, and scaled by the corresponding singular value.

Analyzing the SVD of a BF-CNN on natural images reveals that most singular values are close to zero (Figure 1a). The network is thus discarding all but a very low-dimensional portion of the input image. We can measure an "effective dimensionality", d, of



(a) (b) (c)

Figure 1: **Analysis of the SVD of the Jacobian of a BF-CNN. (a)** Singular value distributions for ten example images, corrupted by noise of standard deviation $\sigma = 50$. For all images, a large proportion of the values are near zero, indicating (approximately) a projection onto a subspace (the *signal subspace*). **(b)** Effective dimensionality of the signal subspaces (sum of squared singular values) as a function of noise level (standard deviation, σ). For comparison, the total dimensionality of the space is 1600. Average dimensionality (red curve) falls approximately as the inverse of σ (dashed curve). **(c)** Histogram of dot products (cosine of angle) between the left and right singular vectors that lie within the signal subspaces.

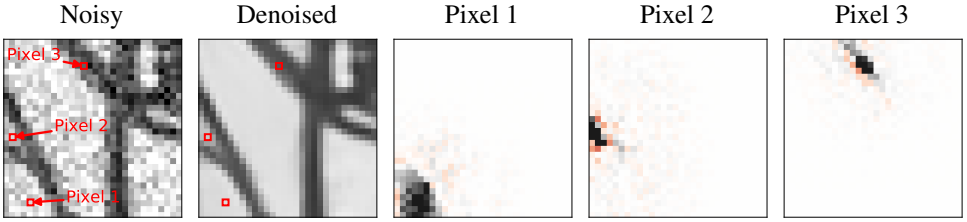


Figure 2: Visualization of the linear weighting functions (rows of A_y in equation 2) of a BF-CNN for three example pixels of an input image. The three rightmost images show the weighting functions used to compute each of the indicated pixels (red squares). All weighting functions sum to one, and thus compute a local average (note that some weights are negative, indicated in red). Their shapes vary substantially, and are adapted to the underlying image content.

this preserved subspace by computing the total noise variance remaining in the denoised image, $f_{\text{BF}}(y)$, which corresponds to the sum of the squares of singular values.

$$\mathbb{E}_n||A_y n||^2 = \mathbb{E}_n||U_y S_y V_y^T n||^2 = \mathbb{E}_n||S_y n||^2 = \mathbb{E}_n \sum_{i=1}^{N} s_i^2 n_i^2 = \sum_{i=1}^{N} \mathbb{E}_n(s_i^2 n_i^2) \approx \sigma^2 \sum_{i=1}^{N} s_i^2 = \sigma^2 d \tag{3}$$

where $d := \mathbb{E}_n||A_y n||^2/\sigma^2 = \sum_{i=1}^{N} s_i^2$. We also observe that the left and right singular vectors corresponding to the singular values with non-negligible amplitudes are approximately the same (Figure 1c). This means that the Jacobian is (approximately) symmetric, and we can interpret the action of the network as projecting the noisy signal onto a low-dimensional subspace, as is done in wavelet thresholding schemes.

For inputs of the form $y := x + n$, the subspace spanned by the singular vectors corresponding to the non-negligible singular values contains x almost entirely, in the sense that projecting x onto the subspace preserves most of its norm. The low-dimensional subspace encoded by the Jacobian is therefore tailored to the input image. This is confirmed by visualizing the singular vectors as images. The singular vectors corresponding to non-negligible singular values capture features of the input image; the ones corresponding to near-zero singular values are unstructured (Figure 3). BF-CNN therefore implements an approximate projection onto an adaptive *signal subspace* that preserves image structure, while suppressing much of the noise.

The signal subspace depends on the noise level. We find that for a given clean image corrupted by noise, the effective dimensionality of the signal subspace decreases as the noise level increases (Figure 1b). At lower noise levels the network detects a richer set of image features, that lie in a larger signal subspace. In addition, these signal subspaces are nested: subspaces corresponding to lower noise levels contain at least 95% of the subspaces corresponding to higher noise levels.

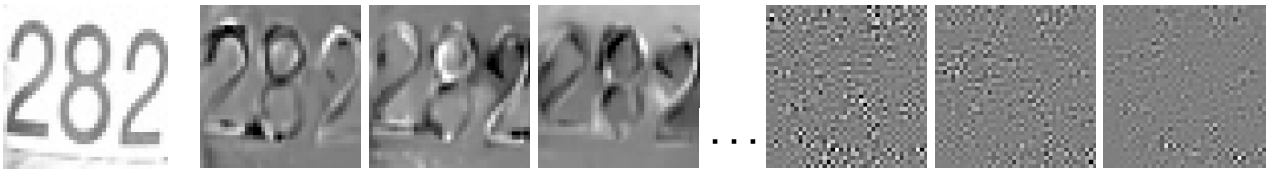
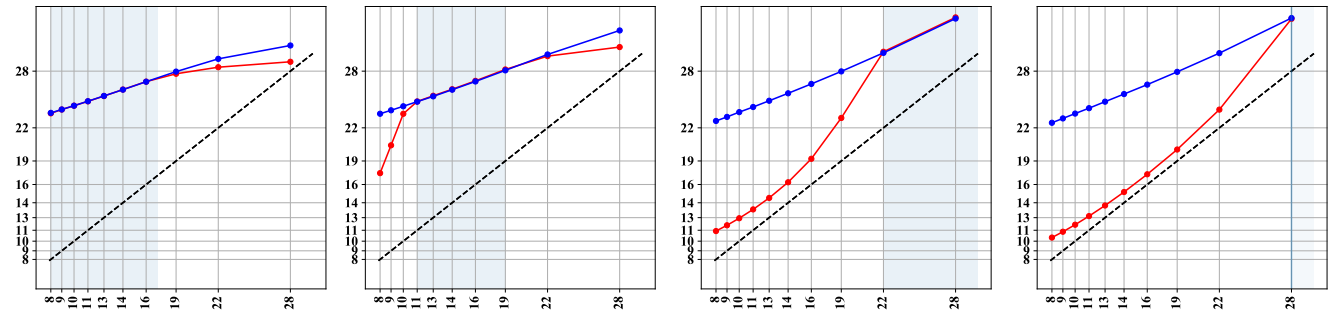


Figure 3: **Visualization of left singular vectors of the Jacobian of a BF-CNN.** The left column shows the original (clean) image. The next three columns show singular vectors of the Jacobian (for the image corrupted by noise with standard deviation $\sigma = 50$) corresponding to non-negligible singular values, which can be seen to capture features from the clean image. The last three columns on the right show singular vectors whose singular values are close to zero: These vectors are noisy and unstructured.



(a) (b) (c) (d)

Figure 4: **Generalization of performance to noise levels outside of the training range.** Performance, quantified as PSNR of the denoised image, as a function of input PSNR, of DnCNN (14) (red curves) and BF-CNN (blue curves). The black dashed line depicts the identity function, for comparison. **(a-c)** The two networks are trained over a fixed ranges of noise levels indicated by a gray background. In all cases, the performance of DnCNN degrades significantly beyond the training range. In contrast, BF-CNN generalizes robustly, as predicted by equation **??**. **(d)** Superposition of results from networks trained on three different low-noise ranges. In all cases, BF-CNN generalizes well, in stark contrast to DnCNN.

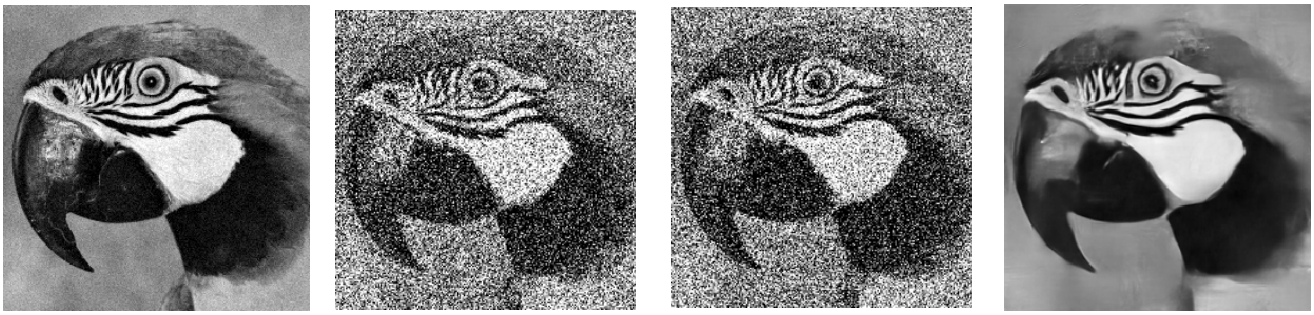
3 Generalization across noise levels

The empirical result that dimensionality is equal to $\frac{\alpha}{\sigma}$, combined with the observation that the signal subspace contains the clean image, explains the observed denoising performance across different noise levels (Figure 4). Specifically, if we assume $A_y x \approx x$, the mean squared error is proportional to σ:

$$\begin{aligned} \text{MSE} &= E_n ||A_y(x+n) - x||^2 \\ &\approx E_n ||A_y n||^2 \\ &\approx \sigma^2 d \\ &\approx \alpha\, \sigma \end{aligned} \tag{4}$$

The scaling of MSE with the square root of the noise variance implies that the PSNR of the denoised image should be a linear function of the input PSNR, with a slope of $1/2$. This provides an empirical target for generalization beyond training range.

We investigate generalization across noise levels, comparing networks with and without net bias. We implement BF-CNNs based on several Denoising CNNs (14; 7; 11; 15). These architectures include popular features of existing neural-network techniques in image processing: recurrence, multiscale filters, and skip connections. To construct BF-CNNs, we remove all sources of additive bias, including the mean parameter of the batch-normalization in every layer (note however that the rescaling parameters are preserved). We train the networks, following the training scheme described in (14), using images corrupted by i.i.d. Gaussian noise with a range of standard deviations. This range is the *training range* of the network. We then evaluate the networks for noise levels that are both within and beyond the training range. Figure 4 compares DnCNN from (14) and its equivalent BF-CNN for different noise levels, inside and outside of the training range. In all cases, DnCNN generalizes very poorly to noise levels outside the training range. In contrast, BF-CNN generalizes robustly, as predicted with a slope of 1/2, even when trained only on modest levels of noise ($\sigma = [0, 10]$). Figure 5 shows an example that demonstrates visually the striking difference in generalization performance. We found that the same holds for the other architectures.



Noisy training image, $\sigma = 10$ (max level) | Noisy test image, $\sigma = 90$ | Test image, denoised by CNN | Test image, denoised by BF-CNN

Figure 5: Denoising of an example natural image by a CNN and its bias-free counterpart (BF-CNN), both trained over noise levels in the range $\sigma \in [0, 10]$ (image intensities are in the range $[0, 255]$). The CNN performs poorly at high noise levels ($\sigma = 90$, far beyond the training range), whereas BF-CNN performs at state-of-the-art levels. The CNN used for this example is DnCNN (14); using alternative architectures yields similar results.

References

[1] CHANG, S. G., YU, B., AND VETTERLI, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing 9*, 9 (2000), 1532–1546.

[2] CHEN, Y., AND POCK, T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Patt. Analysis and Machine Intelligence 39*, 6 (2017), 1256–1272.

[3] DABOV, K., FOI, A., KATKOVNIK, V., AND EGIAZARIAN, K. Image denoising with block-matching and 3d filtering. In *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning* (2006), vol. 6064, International Society for Optics and Photonics, p. 606414.

[4] DONOHO, D., AND JOHNSTONE, I. Adapting to unknown smoothness via wavelet shrinkage. *J American Stat Assoc 90*, 432 (December 1995).

[5] ELAD, M., AND AHARON, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. on Image processing 15*, 12 (2006), 3736–3745.

[6] HEL-OR, Y., AND SHAKED, D. A discriminative approach for wavelet denoising. *IEEE Trans. Image Processing* (2008).

[7] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. Densely connected convolutional networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2017), pp. 4700–4708.

[8] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature 521*, 7553 (2015), 436.

[9] LEFKIMMIATIS, S. Universal denoising networks: a novel cnn architecture for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 3204–3213.

[10] RAPHAN, M., AND SIMONCELLI, E. P. Optimal denoising in redundant representations. *IEEE Trans Image Processing 17*, 8 (Aug 2008), 1342–1352.

[11] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241.

[12] SIMONCELLI, E. P., AND ADELSON, E. H. Noise removal via Bayesian wavelet coring. In *Proc 3rd IEEE Int'l Conf on Image Proc* (Lausanne, Sep 16-19 1996), vol. I, IEEE Sig Proc Society, pp. 379–382.

[13] WIENER, N. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. Technology Press, 1950.

[14] ZHANG, K., ZUO, W., CHEN, Y., MENG, D., AND ZHANG, L. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Processing 26*, 7 (2017), 3142–3155.

[15] ZHANG, X., LU, Y., LIU, J., AND DONG, B. Dynamically unfolding recurrent restorer: A moving endpoint control method for image restoration. *arXiv preprint arXiv:1805.07709* (2018).