How Do Neural Networks Denoise Natural Images?

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

Results presented in the report are joint work with Zahra Kadkhodaie, Prof. Eero P Simoncelli and Prof. Carlos Fernandez-Granda

1 Introduction

Our measurement devices are often imperfect. The microphone of our cell phone picks up background sounds and the images captured from our camera are corrupted by random voltage fluctuations in the light sensors. As in these examples, addition of noise is a common form of corruption that occurs in our measurement devices. Reducing the noise and recovering the signal of interest from corrupted measurements is called *denoising*. To denoise effectively, one has to tell apart signal from the noise. This means that the denoiser should have a good understanding of what the signal should look like. Particularly, for our example with the camera, the denoiser should be able to describe what the image of naturally occurring objects (like people, trees, animals, buildings etc), called natural images, look like. Images of nanoparticles taken with an electron microscope or images acquired from an MRI machine are not classified as natural images - they have very different statistics. We will focus on denoising natural images in this work (Refer to fig 1 for a visual example). Natural images are difficult to describe mathematically or algorithmically. This makes denoising problem a test on how well we understand natural images, in addition to being an important application in our image acquisition systems.

Methods to work with natural images are an active area of research and in the past decade, a specific machine learning technique called convolutional neural networks (CNN) (LeCun et al., 2015), introduced in section 3, has emerged as the defacto standard to deal with image data. CNN based denoisers implement a denoising strategy that is completely data driven. The denoiser is shown thousands of pairs of noisy and clean images which it uses to *learn* a strategy to denoise a new noisy image. This is in stark contrast to traditional methods where denoising algorithms are explicitly designed using known properties of natural images. However, CNN based denosiers have outperformed traditional signal processing denoising algorithms by large margins (Zhang et al., 2017; Chen & Pock, 2017). This superior performance achieved CNNs comes at a cost of interpretability - we don't understand how CNNs denoise images. On a high level, the goal of this work is to advance our understanding of how CNNs denoise natural images and gain intuition and a formal understanding of the denoising mechanism they implement. This understanding can help us better realize the potential and limitations of the current CNN based denoisers.

In the rest of the manuscript, we will develop a detailed understanding of CNN based image denoisers. Section 2 formalizes the denoising problem and introduces some traditional signal processing based denoisers. Section 3 reviews CNNs and CNN based denoisers. Section 4 introduces my previous results from Mohan et al. (2019) showing that CNNs denoise by projecting to an low-dimensional space. Section 5 discusses my current research on how CNNs find this low-dimensional space.

2 Denoising Problem and Classical Solutions

An $m \times n$ image or a natural image with $N = mn$ pixels is equivalent to flattened vector with N entries. That is, our signal x is in $\mathbb{R}^N$. The measurement y that we observe is our signal x corrupted with an additive Gaussian noise $\eta \in \mathbb{R}^N$:

$$y = x + \eta, \text{ where } \eta \sim \mathcal{N}(0, \sigma^2 I_N), \tag{1}$$

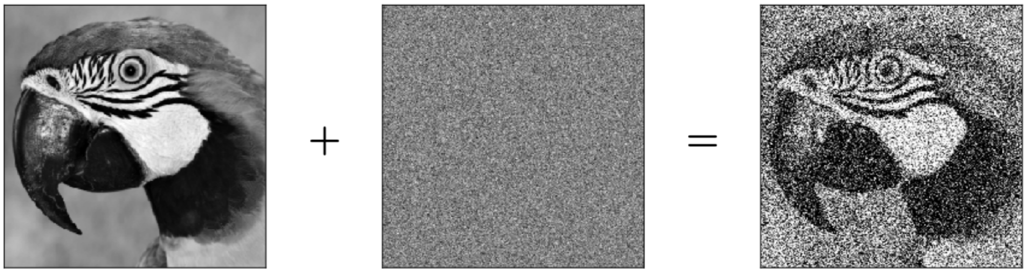


Figure 1: Visual example of denosing problem. The goal of denoising problem is to estimate the clean image from observed noisy image. In this example, the observed noisy image (right) is a sum of a clean image (left) with independent Gaussian noise (middle).

where I_N is an $N \times N$ identity matrix and $\sigma^2 > 0$ denotes the variance of the each dimension of the Gaussian random variable. (Refer to figure 1 for an illustration of the corruption model.)

Solving the denoising problem involves finding a function $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$ such that a noisy observation y can be mapped to a good estimate of x, i.e $f(y) \approx x$. We say that $f(y)$ is a good denoised image if the squared error $||x - f(y)||_2^2$ between the clean image x and the denoised image $f(y)$ is small. We can assume that signal x and noisy observation y are realizatiosn of a random variable $\mathcal{X}$ $\mathcal{Y}$ respectively. The denoising function that is optimal across all realizations of $\mathcal{X}$ and $\mathcal{Y}$ is defined as the function f that minimizes the squared error:

$$f_{\text{opt}} := \arg\min_f E_{\mathcal{X}\times\mathcal{Y}} ||x - f(y)||_2^2, \tag{2}$$

where the expectation E is taken over the joint distribution of clean and noisy images $\mathcal{X} \times \mathcal{Y}$.

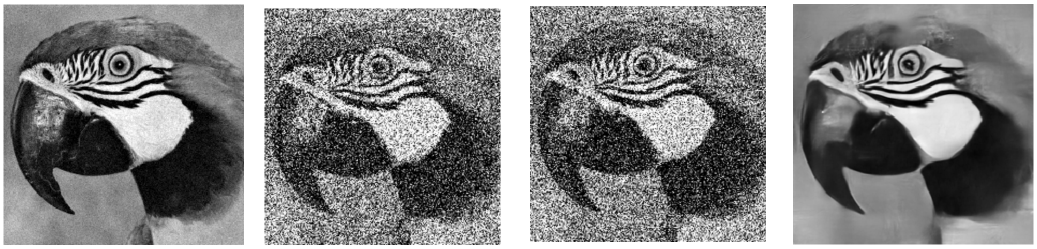
If the noise level (or noise standard deviation) σ is not known to us and the denoiser is expected to work for a range of noise levels, then the problem is called blind denoising. In blind denoising, we assume that σ is uniformly distributed between 0 and a maximum value σ_{max}. This will change the distribution of $\mathcal{Y}$ to include the change in noise levels and the expectation in equation 2 will implicitly be taken over σ as well. If we do not make assumptions on the distribution of $\mathcal{X}$, then we approximate the expectation in 2 by the empirical expectation over a dataset of noisy and clean image pairs $\{(y_i, x_i)\}_{i=1}^n$:

$$\hat{f}_{\text{opt}} := \arg\min_f \frac{1}{n} \sum_{i=1}^{n} ||x_i - f(y_i)||_2^2, \tag{3}$$

where n is the number of examples in our dataset.

Perhaps the simplest solution for a denoising problem is the Wiener filter, where the optimization problem in equation 2 is simplified by constraining f and $\mathcal{X}$. The Wiener filter assumes that the signal $\mathcal{X}$ is translation invariant and follows a Gaussian distribution and the denoised image $f(y)$ is a linear function of the noisy image y f (Wiener, 1950). The Wiener filter computes the denoised value of a particular noisy pixel as a weighted average of the noisy pixel values in its neighbourhood. The neighbourhood over which the Wiener filter averages increases with increasing noise level, as we would expect. The main limitation of Wiener filtering is that it smooths the edges in images and thus does not produce aesthetically pleasing images.

The smoothing of edges in the Wiener filter is because its linear nature - the same filter is applied everywhere in the image without adapting to the underlying image content. To adapt to the image content, the denoising function f has to be nonlinear. The central idea behind many nonlinear denoisers is to find a smart scheme to separate noise from the signal. This separation is difficult to achieve in the pixel domain, but can be achieved with a transformation to a carefully chosen space. In this transformed space, often, small coefficient values correspond to noise and large values corresponds to signal and thus, noise can be separated from signal using nonlinear thresholding operation. The thresholded coefficients can be transformed back to the pixel space using the inverse transformation (Donoho & Johnstone, 1995; Simoncelli & Adelson, 1996; Chang et al., 2000). From a linear algebraic prospective, the equivalent operation implemented by such a denosier can be



Noisy training image, $\sigma = 10$ (max level)

Noisy test image, $\sigma = 90$

Test image, denoised by CNN

Test image, denoised by BF-CNN

Figure 2: Denoising performance of a CNN and a BF-CNN when tested on a noisy image far outisde its training range. Both the CNN and BF-CNN were trained on $\sigma \in [0, 10]$ and were tested on a noisy image with $\sigma = 90$, which is far outside the training range. BF-CNN is able to achieve state-of-the-art performance in this task, while CNN barely denoises the image. Image reproduced from Mohan et al. (2019).

thought of as a projecting noisy input to a lower-dimensional manifold containing the actual signal. In section 4 we will see that modern deep learning methods also denoise images by implementing a similar projection based mechanism.

3 Convolutional Neural Network Based Denoising

For a noisy input image $y \in \mathbb{R}^N$ with N pixels, the denoising function $f : \mathbb{R}^N \to \mathbb{R}^N$ computed by a neural network is

$$f(y) = W_L R(W_{L-1} ... R(W_1 y + b_1) + ... + b_{L-1}) + b_L, \tag{4}$$

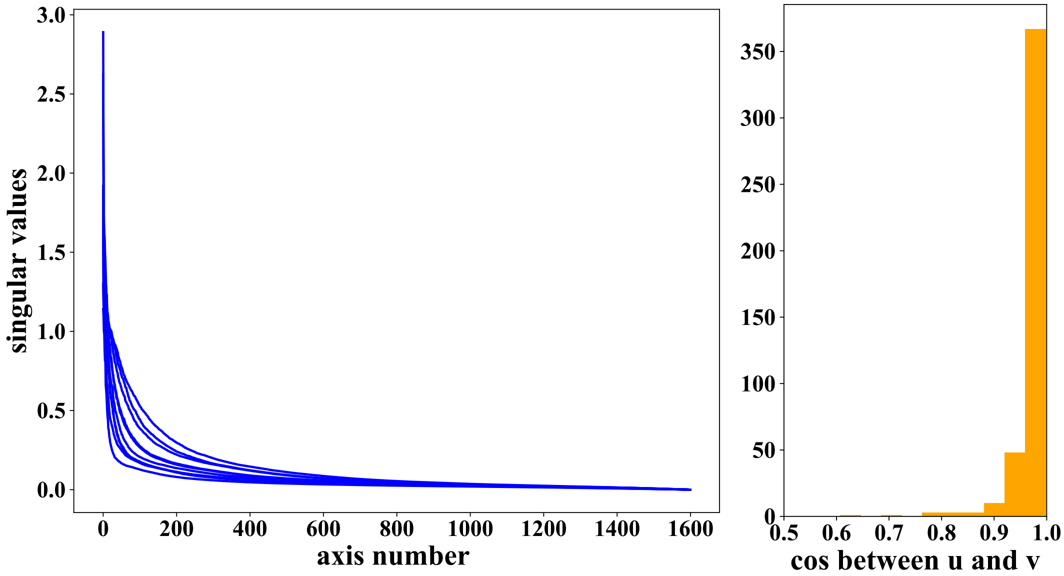
where W_i is called the weight matrix and b_i is bias vectors at layer i. The function $R(z) = \max(0, z)$ represents a rectified linear unit (ReLU) non-linearity. ReLU is applied element wise to any tensor input. If the neural network is convolutional, W_is are restricted to be convolutional matrices. The process of finding the optimal parameters W_is and b_is by solving the optimization problem in equation 3 is called *training*. Neural networks that are trained on large databases of noisy and clean natural images to minimize mean square error achieves current state-of-the-art denoising performance (Zhang et al., 2017; Huang et al., 2017; Ronneberger et al., 2015; Zhang et al., 2018).

When training a neural network, we need real data to approximate the expectation in 2 and arrive at equation 3. Since the expectation in equation 2 is over the joint distribution of $\mathcal{X} \times \mathcal{Y}$ we needs pairs of noisy and clean images to approximate the expectation. While we have datasets with clean images (Martin et al., 2001), we have very limited data on pairs of noisy and clean images. To overcome this obstacle, we use the clean images from our dataset and use our additive gaussian noise model (equation 1) to generate corresponding noisy images for training. For blind denoising, we train on noisy images generated with range of noise standard deviations σ. The range of noise standard deviations, $[\sigma_{\min}, \sigma_{\max}]$ included in the training images is called the *training range* of denoiser.

A trained CNN based denoiser (parameterized according to equation 4) generalizes well to new images not seen during the training and achieves state-of-the-art performance. However, they fail to denoise well if the test image is corrupted with a noise level not seen during training (refer to figure 2 for a visual example). In Mohan et al. (2019) my co-authors and I showed that this over-fitting to training noise levels is caused by the additive bias terms ($b_1, b_2, \ldots, b_L$ in equation 4) in the network. Further, we showed that removing these additive bias terms and recomputing the weights enabled the network to generalize to unseen noise levels (refer to figure 2 for a visual example). We call the reparameterized CNN with no additive bias terms *Bias Free CNN (BF-CNN)*. For the rest of the manuscript, our analysis is on BF-CNNs.

4 Understanding CNN based Denoisers

The highly nonlinear structure of CNNs makes it difficult to understand its working. However, bias free CNNs exhibit a special structure that we can exploit to provide some interpretation into the



(a)

(b)

Figure 3: Analysis of the SVD of the Jacobian of a BF-CNN for ten natural images, corrupted by noise of standard deviation $\sigma = 50$. **(a)** Shows the singular value curve for the Jacobian computed for 10 different images. The singular values decrease quickly implying that the network is transforming the noisy image to a low dimensional space. **(b)** The histogram of the dot product between left and right singular vectors corresponding to non-negligible singular values. The dot product is very close to 1 implying that the matrix is approximately symmetric. (Image reproducted from (Mohan et al., 2019))

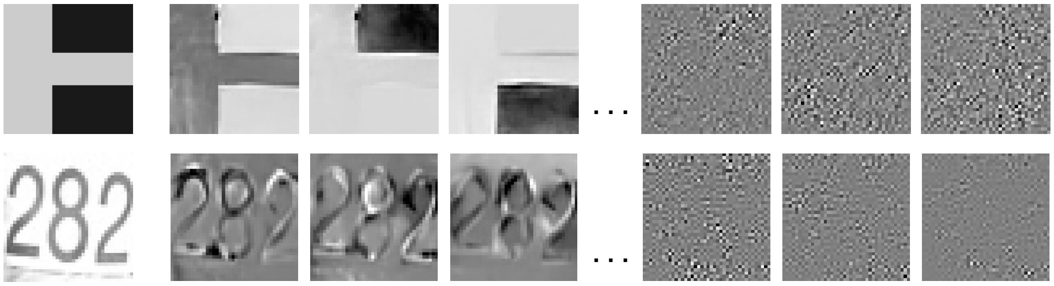


Figure 4: Visualization of the left singular vectors (u_is) for two input noisy images (top row and bottom row). The input images were at noise level $\sigma = 30$. The first colum shows the clean image, the next three column shows singular vectors corresponding to non-negligible singular value and the last three columns shows singular vectors corresponding to very small singular values. Note that the singular vectors corresponding to non-negligible singular values looks like the image while the ones from small singular values looks like noise. (Image reproducted from (Mohan et al., 2019))

working of these networks. For a given noisy image y, the first layer of a bias free network performs a linear transform $W_1 y$ followed by a ReLU $R(W_1 y) = \max(0, W_1 y)$. The ReLU function sets negative entries in the input to zero and leaves the positive entries alone. Therefore, for a given input to the ReLU, the action of ReLU can be thought of as multiplication by a diagonal matrix with 1 corresponding to the positives entries of the input and 0 for the negative entries of input. Hence, the entire transformation implemented by the first layer $R(W_1 y)$ is linear for a given input y. Subsequent layers apply the same transformation again and again and thus the entire function transformation by a bias free network is a cascade of linear operations for a given input y. That is,

$$f_{\text{BF}}(y) = W_L R(W_{L-1} ... R(W_1 y)) = A_y y, \tag{5}$$

where A_y is the Jacobian of $f_{\text{BF}}(\cdot)$ evaluated at y. The subscript on A_y serve as a reminder that this linear representation depends on the input y and changes for every input.

The action of a ReLU function on a vector $z \in \mathbb{R}^N$ can be characterized by a binary vector v_z of size N with entries 0 and 1. If v_z has 1 in the dimension where z is positive and 0 where z is negative, then $R(z) = v_z \odot z$ where $\odot$ represents the element wise product operator. Extending this, the transformation implemented by the bias free network in equation 5 can be completely characterized by $(L-1)$ binary vectors corresponding to the $(L-1)$ ReLU functions in the networks. For the points in a small region in the input space the binary vector activation patterns of all the ReLUs remains the same and hence the linear representation remains the same. Thus, given an input noisy image y the denoising function $f(z) = A_y z$ for all the points z in a neighbourhood of y where the ReLU activation remains the same. Therefore, the denoising map can be completely characterized by the Jacobian matrix A_y for a local neighbourhood around y.

This characterization of a highly non-linear neural network by its linear form (or the Jacobian) allows us analyze the local denoising map using linear algebraic tools. For a given noisy image y, we can analyze the properties of the denoising map around y using the SVD of the Jacobian matrix: $A_y = USV^T$, where U and V are orthonormal matrices which span the column space and row space respectively of A_y and S is a diagonal matrix with singular values. We can express the action of the network as follows:

$$f_{\text{BF}}(y) = A_y y = USV^T y = \sum_{i=1}^{N} s_i (V_i^T y) U_i. \tag{6}$$

The denoised output $f_{\text{BF}}(y)$ lives in a subspace spanned by the vectors $U_1, U_2, \ldots, U_N$.

Our analysis of the SVD of Jacobian matrix over a set of natural images shows that most singular values are very close to zero (Figure 3a). Therefore, $f_{\text{BF}}(y)$ maps to a low dimensional space. Further, the dot product, $u_i^T v_i \approx 1$ when s_i is non-negligible indicating that $u_i \approx v_i$ whenever s_i is non-negligible (Figure 3b). Therefore, the Jacobain matrix A_y is approximately symmetric. Since A_y is symmetric, low rank and has a largest singular value around 1, we can interpret the action of the network as approximately projection of the noisy signal onto a low-dimensional space. We verify that the low-dimensional space depends on the signal by visualizing the basis vectors of this space (Figure 4) (Mohan et al., 2019).

5 Future Work: How does the network find this subspace?

In section 4 we showed that the action of CNN based denoiser can be interpreted as a projection to a low dimensional space. The projection matrix or the Jacobian A_y, is composed of weight matrices of the network and binary activation patterns of ReLU. The binary activation patterns of ReLU depends on the input and hence the projection matrix or the space changes for each input. Future work includes exploring questions on how the network finds this space. Specifically:

- Why does this cascade of weight matrices and ReLU activation patterns results in projection matrices?
- How does rows of different weight matrices combine to produce the final projection matrix? What is encoded in each of these weight matrices?

Answering these questions will help us better understand the denoising mechanism implemented by neural networks. This understanding will enable us to design faster and more robust networks.

REFERENCES

S Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9(9):1532–1546, 2000.

Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Patt. Analysis and Machine Intelligence*, 39(6): 1256–1272, 2017.

D Donoho and I Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J American Stat Assoc*, 90(432), December 1995.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.

Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. *arXiv preprint arXiv:1906.05478*, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

E P Simoncelli and E H Adelson. Noise removal via Bayesian wavelet coring. In *Proc 3rd IEEE Int'l Conf on Image Proc*, volume I, pp. 379–382, Lausanne, Sep 16-19 1996. IEEE Sig Proc Society. doi: 10.1109/ICIP.1996.559512.

Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. Technology Press, 1950.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Processing*, 26(7): 3142–3155, 2017.

Xiaoshuai Zhang, Yiping Lu, Jiaying Liu, and Bin Dong. Dynamically unfolding recurrent restorer: A moving endpoint control method for image restoration. *arXiv preprint arXiv:1805.07709*, 2018.