CDLNet: Robust and Interpretable Denoising through Deep Convolutional Dictionary Learning

Nikola Janjušević, Amirhossein Khalilian-Gourtani, Yao Wang

Abstract—Deep learning based methods hold state-of-the-art results in image denoising, but remain difficult to interpret due to their construction from poorly understood building blocks such as batch-normalization, residual learning, and feature domain processing. Unrolled optimization networks propose an interpretable alternative to constructing deep neural networks by deriving their architecture from classical iterative optimization methods, without use of tricks from the standard deep learning tool-box. So far, such methods have demonstrated performance close to that of state-of-the-art models while using their interpretable construction to achieve a comparably low learned parameter count. In this work, we propose an unrolled convolutional dictionary learning network (CDLNet) and demonstrate its competitive denoising performance in both low and high parameter count regimes. Specifically, we show that the proposed model outperforms the state-of-the-art denoising models when scaled to similar parameter count. In addition, we leverage the model’s interpretable construction to propose an augmentation of the network’s thresholds that enables state-of-the-art blind denoising performance and near-perfect generalization on noise levels unseen during training.

Index Terms—dictionary learning, blind denoising, interpretable deep learning

I. INTRODUCTION AND BACKGROUND

In recent years, deep-learning methods have become state-of-the-art in imaging inverse problems such as denoising [18], [19], compressed-sensing [10], super-resolution [18], and more. However, these deep neural networks (DNN) and their building blocks are currently not well understood, making it ineffective to draw on knowledge from decades of signal processing and optimization research for insight or improvement. As a result, poorly understood tools such as batch-normalization [9], residual-learning, and feature-domain processing, have become staples for deep-learning architectures. So called “unrolled optimization” DNNs [15] propose a promising avenue to remedy this by deriving their network architecture from $K$ iterations of an iterative optimization algorithm. Existing unrolled optimization DNNs are able to achieve state-of-the-art results on image restoration tasks without the use of the previously mentioned standard deep-learning toolbox [10], [16]. However, these formulations mostly leverage their interpretability to reduce network parameter count in comparison to popular DNNs. In this work, we present a further desirable quality to be gained from interpretable model construction: robustness.

Our proposed method tackles natural-image denoising via unrolled convolutional dictionary learning (CDL), which we call CDLNet. The convolutional dictionary learning model is rooted in the sparse representation prior, where we assume our signals $x \in \mathbb{R}^N$ may be represented by a linear combination of only a few vectors (atoms) from a larger dictionary, $D$,

$$\exists \ z \ s.t. \ x = Dz, \ ||z||_0 \ll N, \quad (1)$$

where $||z||_0 := |\{i : z[i] \neq 0\}|$ is the $\ell_0$ pseudo-norm (a.k.a counting norm). With some assumptions on the level of sparsity in $z$ and diversity (incoherence) of the dictionary atoms, stable and unique recovery of $z$ from $x$ may be guaranteed [12]. A primary focus of signal processing literature in the last decade and a half has been on the selection of a representation dictionary, in which learning based methods such as KSVD [1] brought state-of-the-art performance in image restoration tasks. Given a data-set $\{x_i\}_{i=1}^n$, the dictionary learning problem may be formulated with an $\ell_1$-norm relaxation of the $\ell_0$ pseudo-norm, which has been shown to promote sparsity,

$$\min_{D \in \mathcal{C}(z_i)} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} ||Dz_i - x_i||^2_2 + \lambda ||z_i||_1. \quad (2)$$

The regularization parameter $\lambda$ gives a data-fidelity vs. sparsity trade-off, and $D$ is restricted to the constraint set of columns in the unit-ball,

$$\mathcal{C} = \{D : ||d^j||^2_2 \leq 1 \forall j\}, \quad (3)$$

to prevent arbitrary scaling of coefficients. This formulation is amenable to the denoising problem, $y = x + n$, by replacing the data fidelity term with $||Dz_i - y_i||^2_2$. In this work, we consider the denoising problem with additive white Gaussian noise (AWGN), where $n \sim N(0, \sigma_n^2 I)$.

The problem in (2) is non-convex and thus commonly solved via a Gauss-Seidel split, alternating between sparse-coding with a fixed dictionary, and updating the dictionary given sparse codes [1]. The iterative soft thresholding algorithm (ISTA) is one such sparse-coding algorithm, whose iterates are defined as,

$$z^{(k+1)} = ST\left(\frac{z^{(k)} - \eta^{(k)} D^\top (Dz^{(k)} - y)}{\eta^{(k)} + \lambda}\right), \quad (4)$$

where $ST(x, \tau)[i] := \text{sign}(x[i]) \max(0, |x[i]| - \tau)$ is the element-wise soft-thresholding operator with threshold $\tau \geq 0$, $\eta^{(k)}$ is a step-size parameter, and $\lambda$ is the sparsity penalty from (2). In the case of a convolutional dictionary, we use $M$ filters $d^j \in \mathbb{R}^p$ whose integer translates form the columns of $D$. The application of the dictionary (synthesis convolution) is given

$$d^j \star x = \sum_{l} d^j[l] x[l].$$

arXiv:2103.04779v1 [eess.IV] 5 Mar 2021
by $Dz = \sum_{j=1}^M d^j \ast z^j$, where $z^j$ is the $j$-th channel of $z$. The corresponding analysis convolution is defined channel-wise by $(D^\top x)^j = d^j \ast x$ where $d^j$ denotes the reversal of the filter. In the case of 2D input signals $x \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$, we consider 2D square filters $d^j \in \mathbb{R}^{\sqrt{p} \times \sqrt{p}}$.

Making a connection between ISTA and DNNs is tempting due to their iterative use of linear operators followed by pointwise non-linearities. Our work builds off the seminal work of [5] by employing a learned ISTA encoder followed by a linear synthesis dictionary.

II. RELATED WORKS

The authors of [17] proposed an unrolled CDL DNN for image denoising and image inpainting, demonstrating results competitive with that of KSVD [11] in a fraction of the computational time. Our baseline proposed method builds off of their framework, introducing strided convolutions, preprocessing, and weight initialization as key steps to reaching denoising performance on par with state-of-the-art deep learning methods.

In [16], an argument is given suggesting the ill-conditioned nature of the convolutional sparse-coding (CSC) model in the representation of natural images. Their proposed model, CSCNet, uses large strides on the order of the filter size, along with averaging reconstructions from shifted input signals – effectively returning to a patch-based approach. CSCNet demonstrates an alternative approach to CSCNet that is not inconsistent with their analysis, achieving superior performance on image denoising benchmarks without the use of “shift-averaging”.

The authors of [14] showed that several DNN denoising models, such as DnCNN [18], exhibit a catastrophic failure in denoising performance when presented with input noise-levels outside of the model’s training range. They proposed “bias-free” versions of the networks (ex. BF-DnCNN), which were demonstrated to posses the desired noise-level generalization property. Our work provides an alternate route for bringing robustness to DNN denoising by adapting the network’s thresholds with the input noise-level.

The authors of [7] replace the ReLU non-linearities of DnCNN [18] with soft-thresholding operators whose thresholds are proportional to the input signal standard-deviation. Our noise adaptive model instead employs soft-thresholding operators whose thresholds achieve state-of-the-art blind denoising performance and allow the model to generalize above and below training noise-levels (Section IV-D).

III. PROPOSED METHODOLOGY

A. The CDLNet architecture

In this section we introduce the convolutional dictionary learning network (CDLNet) for natural image denoising. The architecture involves a convolutional learned ISTA encoder with $K$ unrollings and a convolutional synthesis dictionary, $D$.

$$x = Dz^{(K)}, \quad z^{(0)} = 0, \quad k = 0, 1, \ldots, K - 1,$$

$$z^{(k+1)} = ST\left(\alpha^{(k)}_j \left(B^{(k)}z^{(k)} - y\right), \tau^{(k)}\right),$$

where $\Theta = \{\alpha^{(k)}_j, B^{(k)}_j, \tau^{(k)}_j\}^{K-1}_{k=0}, D$ are the set of learned parameters. $B, D$ are all $M$-channel convolutional synthesis operators, and $A^\top$ are $M$-channel convolutional analysis operators. The non-negative thresholds $\tau^{(k)}_j \in \mathbb{R}^M$ are subband dependent, corresponding to a learned weighted $\ell_1$ norm in [2].

The output denoised image is given by $\hat{x} = Dz^{(K)}$. The block diagram of the proposed CDLNet is given in Fig. 1.

The convolution analysis and synthesis operators as shown are highly redundant transformations, increasing the number of coefficients from $N$ to $MN$ in analysis. This level of redundancy can become computationally burdensome as the number of filters ($M$) increases. To mitigate this, we can replace each of the convolution synthesis operators $A, B, D$ represented by $W$, with a sub-sampled version $W\Delta_s$, where $\Delta_s$ is the zero-filling operator with stride $s$. Note that $\Delta_s^\top$ is the sub-sampling operator with stride $s$. For 2D images we consider $\Delta_s$ with stride $s$ in both horizontal and vertical directions. The computational complexity of this network is roughly $O(KMNp/s^2)$ for input signals in $\mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ and filters of size $\sqrt{p} \times \sqrt{p}$. Replacing the convolution synthesis operators $A, B, D$ with their sub-sampled versions allows for the possibility of learning more diverse filters while keeping the computation under control.

B. Noise-adaptive thresholds

We leverage CDLNet’s signal-processing and optimization derivation to provide adaptation to varying input noise-levels in a single model. We augment the learned thresholds in $\tau$ to be proportional to the input noise standard deviation ($\sigma_n$),

$$\tau^{(k)} = \lambda^{(k)} \sigma_n, \quad k = 0, 1, \ldots, K - 1$$

where $\lambda^{(k)} \in \mathbb{R}^M$ is a learned parameter. We propose this augmentation of the thresholds based on the “universal threshold theorem” [12], from the Wavelet denoising literature, $\tau = \sigma_n \sqrt{2 \log \sqrt{N}}$. This can be shown to minimize the
upper bound of the risk function of an element-wise denoising operator \([12]\). With this augmentation, we interpret the proportionality constant \(\lambda^{(k)}\) as learning the gain factor between the image domain noise-level \(\sigma_n\) and the \(i\)-th channel’s noise-level at layer \(k\). This framework has the added benefit of decoupling noise-level estimation from denoising, allowing for trade-off between exact estimation and speed at inference time. We explore this trade-off using two different noise-level estimation algorithms at inference time, in Section [IV-D]. We refer to CDLNet with the above noise-adaptive augmentation as CDLNet-A.

IV. EXPERIMENTS

A. Training and Inference

Models: are trained by the Adam \([9]\) optimizer on the \(\ell_2\)-loss with parameter constraints,

\[
\text{minimize} \quad \|x - \hat{x}(y; \Theta)\|_2^2, \tag{7}
\]

where \(C\) is the unit-ball constraint \([3]\) imposed on all convolution operators. Constraints are enforced by projection after each gradient descent step. Filters of size \(7 \times 7\) are used. We denote the models with \(M=32, K=20\), when trained on a single noise level, as CDLNet-S, and when trained over a noise range with and without adaptive thresholds as CDLNet-A and CDLNet-B, respectively. Similarly, we denote the models with \(M=169, K=30\) as Big-CDLNet. Unless specified otherwise, CDLNet models use stride 1 convolutions and Big-CDLNet models use a stride of 2. Our framework is implemented in PyTorch and provided online \([8]\).

Dataset: All CDLNet models and variants are trained on the BSD432 dataset \([13]\). Input signals are preprocessed with division by 255, random crops of \(128 \times 128\), random flips and rotations, and mean-subtraction. Models trained across noise-levels are done so by uniform sampling of \(\sigma_n \in \sigma_n^{\text{train}}\).

Training: A mini-batch size of 10 samples is used. An initial learning rate of \(1e-3\) is used, and reduced by a factor of 0.95 every 50 epochs for a maximum of 6000 epochs or until convergence. Similar to the method in \([10]\), we backtrack our model to the nearest checkpoint upon divergence, reducing the learning rate by a factor of 0.8.

Initialization: We initialize all convolution operators with the same weights drawn from a standard normal distribution. This comes from the intuition that the majority of our learned channels will consist of band-pass signals modeling image texture, and so our filters should be zero-mean. Following \([16]\), as an initialization step, we normalize the convolution operators by their spectral norm in correspondence with the maximum uniform step-size of ISTA.

**Noise-level estimation**: We employ two different noise-level estimation algorithms for blind denoising: one based on the median absolute deviation (MAD) of the input’s diagonal wavelet coefficients \([2]\), and the other based on the principal component analysis of a subset of the input’s patches (PCA) \([11]\). These estimators offer two ends of the trade-off between speed (MAD) and accuracy (PCA).

Experiments were conducted with an Intel Xeon(R) Platinum 8268 CPU at 2.90GHz, an Nvidia RTX 8000 GPU, and 4GB of RAM, running Linux version 3.10.0.

B. Single noise-level performance

In these experiments, we train models for individual noise levels. Table I shows the results of the CDLNet model in the small parameter count regime (CDLNet-S) against the other leading CDL based DNN, CSCNet \([16]\), and in the large parameter count regime (Big-CDLNet-S) against state-of-the-art denoising models DnCNN \([18]\) and FFDNet \([19]\). CDLNet outperforms CSCNet in much less computational time at inference, without the use of stride, by using fewer filters and untying its weights between unrollings. We are also able to outperform the listed state-of-the-art deep learning methods by scaling our model (Big-CDLNet) to comparable size. Inference timings given show that our larger model’s use of stride yields a manageable computational complexity. The non-learned method BM3D \([3]\) is given as a classical baseline, and its timing is an order of magnitude greater than the learned methods.

**TABLE I**

| \(\sigma\) | BM3D | CSCNet | CDLNet-S | DnCNN | FFDNet | Big-CDLNet-S |
|---|---|---|---|---|---|---|
| 15 | 31.07 | 31.60 | 31.40 | 31.63 | 31.72 | 31.74 |
| 25 | 28.57 | 28.93 | 28.57 | 29.19 | 29.22 | 29.26 |
| 50 | 25.62 | 26.04 | 26.19 | 26.29 | 26.23 | 26.35 |

Parameters: 486K, 556K, 510K. GPU time: 143 ms, 9 ms, 7 ms, 23 ms, 15 ms.
In Table I we use stride 1 convolutions in our small model and stride 2 convolutions for our big model. Table II empirically verifies these choices as optimal by showing the effect of stride on output PSNR, averaged over a gray-scale version of the Kodak dataset [4]. Note that unlike CSCNet we do not employ any “shift-averaging”. We see that the redundancy of the large model allows for use of stride 2 without a denoising performance penalty.

| Stride | CDLNet-S | Big-CDLNet-S |
|--------|----------|--------------|
| 1      | 30.19    | 30.37        |
| 2      | 30.09    | 30.39        |
| 3      | 29.75    | 30.28        |
| 4      | 29.21    | 29.83        |

C. Comparison of learned dictionaries

In this section we compare the learned representations from CDLNet and CSCNet [16]. Fig. 2 shows the convolutional dictionaries obtained from the CDLNet models and CSCNet. We see that Big-CDLNet offers a greater diversity in its learned filters compared to CSCNet. The baseline CDLNet model, despite having a relatively small number of filters, is shown to mostly learned directional “Gabor-like” filters, with some texture components.

D. Blind denoising and generalization across noise-levels

In this section, we consider the blind denoising and generalization scenarios and compare the models equipped with the proposed adaptive threshold schemes to other models. In Fig. 3, we show the performance of the models trained on the noise range \( \sigma_{\text{train}}^{\text{test}} = [15, 35] \) and tested on different noise levels \( \sigma_{\text{test}} = [5, 50] \). Big-CDLNet-A and Big-CDLNet-B refer to the proposed model trained on the training noise range with and without adaptive thresholds, respectively. Additionally, we trained the blind denoising version of DnCNN [18]. DnCNN-B, on \( \sigma_{\text{train}}^{\text{test}} = [15, 35] \), denoted as DnCNN-B\(^*\) in Fig. 3. The single points on the plot in Fig. 3 show the performance of Big-CDLNet-S model with single noise level training (i.e. \( \sigma_{\text{train}}^{\text{test}} = \sigma_{\text{test}} \)).

As shown in Fig. 3, all networks perform closely over the training noise-range. On the other hand, when tested on noise-levels outside the training range, the network with adaptive thresholds (Big-CDLNet-A) greatly outperforms the other models. We observe that models without noise-adaptive thresholds have a very significant performance drop compared to the noise-adaptive model (Big-CDLNet-A) when generalizing above the training noise level, while Big-CDLNet-A nearly matches the performance of the models trained for a specific noise-level (Big-CDLNet-S) across the range. In spite of increasing input signal-to-noise ratio for noise-levels below the training range, we observe that models without noise-adaptive thresholds have diminishing performance returns (note the plateau of Big-CDLNet-B and DnCNN-B\(^*\) in \( \sigma_{\text{test}} = [5, 15] \)). On the other hand, denoising behavior of Big-CDLNet-A extends to the lower noise-range. We observe reduced generalization performance at the very low noise-level range of \( \sigma_{\text{test}} = 5 \). This may be explained by the need for a different thresholding model when the signal variance is much greater than that of the noise [12].
A visual comparison of the denoising generalization of Big-CDLNet-A and DnCNN-B$^*$ is shown in Fig. 4. Our proposed adaptive model provides visually appealing results at the unseen noise-level ($\sigma^*_{\text{test}} = 50$), while DnCNN-B$^*$ fails to generalize and produces unwanted artifacts.

We further compare the blind denoising and generalization capabilities of the proposed method in Table III. As observed in [14], the BF-DnCNN model has reduced performance in the training range compared to DnCNN-B while avoiding the failure outside the training range. Big-CDLNet-A outperforms the DnCNN-B inside the training range and also provides improved generalization outside the range compared to BF-DnCNN. Note that the Big-CDLNet-A still has minor performance drop compared to a single noise level mode (Big-CDLNet-S) but this reduction is less significant than that of BF-DnCNN [14].

| Model         | 5  | 15 | 25 | 50 | 75 |
|---------------|----|----|----|----|----|
| DnCNN-B       | 37.65 | 31.60 | 29.15 | 26.22 | 18.74 |
| BF-DnCNN      | 37.72 | 31.58 | 29.12 | 26.30 | 24.76 |
| Big-CDLNet-A  | 37.73 | 31.62 | 29.20 | 26.30 | 24.76 |

TABLE III

BLIND DE NOISING AND GENERALIZATION COMPARISON TO DnCNN-B [18] AND BF-CNN [14]. ALL MODELS ARE TRAINED ON $\sigma^*_{\text{train}} = [0, 55]$. AVERAGE TEST PSNR ON BSD68 [19] IS REPORTED.

In the blind denoising and generalization experiments of Table III, Figures 5 and 6, we employ the PCA based noise estimation algorithm [11] at inference time. Table IV shows the difference in denoising performance between using the ground-truth (GT) noise-level and the two previously mentioned noise-level estimation algorithms. The PCA based algorithm allows us to attain near ground-truth denoising performance at the cost of increased computation. The wavelet based estimation method (MAD) offers essentially no computational overhead but significantly decreases denoising performance at the lower noise-level ranges.

| Est. Algo. | 5  | 15 | 25 | 50 | 75 |
|------------|----|----|----|----|----|
| GT         | 37.75 | 31.63 | 29.20 | 26.31 | 24.80 |
| PCA [11]   | 37.73 | 31.62 | 29.20 | 26.30 | 24.76 |
| MAD [12]   | 37.18 | 31.55 | 29.18 | 26.30 | 24.75 |

TABLE IV

EFFECT OF NOISE ESTIMATION ALGORITHM ON PERFORMANCE OF THE NOISE ADAPTIVE MODEL BIG-CDLNET-A.

V. CONCLUSION

We have proposed CDLNet as an interpretable deep neural network construction for the image denoising task. CDLNet achieves superior performance to other CDL based networks at a small parameter count, while avoiding the computationally expensive shift-averaging scheme. Additionally, we propose small-strided convolutions to enable the use of an increased model parameter count while retaining a manageable computational complexity. We show that such models perform competitively with state-of-the-art DNNs. Our results demonstrate that many of the popular tools from the deep learning toolbox (batch-norm, residual learning, feature domain processing) may not be necessary for the denoising task. We further leverage the interpretability of our network to propose a noise-level adaptive thresholding scheme that brings state-of-the-art blind denoising performance and robustness to noise-levels mismatch between training and inference.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311–4322, 2006.
[2] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” IEEE Transactions on Image Processing, vol. 9, no. 9, pp. 1532–1546, 2000.
[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.
[4] H. Sreter and R. Giryes, “Learned convolutional sparse coding,” in Advances in Neural Information Processing Systems, 2020, pp. 15 664–15 675.
[5] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” IEEE Signal Processing Magazine, vol. 38, no. 2, pp. 18–44, 2021.
[6] D. Simon and M. Elad, “Rethinking the CSC model for natural images,” in Advances in Neural Information Processing Systems, 2019, pp. 2274–2284.
[7] H. Sreter and R. Giryes, “Learned convolutional sparse coding,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 2191–2195.
[8] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311–4322, 2006.
[9] F. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations, 2015.
[10] K. Isogawa, T. Ida, T. Shiodera, and T. Takeguchi, “Deep shrinkage convolutional neural network for adaptive noise reduction,” IEEE Signal Processing Letters, vol. 25, no. 2, pp. 224–228, 2017.
[11] N. Janjušević, CDLNet repository, 2021. [Online]. Available: https://github.com/nkopj/CDLNet
[12] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations, 2015.
[13] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way. Elsevier Science, 2008.
[14] 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 Proceedings of Eighth IEEE International Conference on Computer Vision, vol. 2, 2001, pp. 416–423.