IMAGE DENOISING WITH GRAPH-CONVOLUTIONAL NEURAL NETWORKS

Diego Valsesia
Politecnico di Torino
Torino, Italy
diego.valsesia@polito.it

Giulia Fracastoro
Politecnico di Torino
Torino, Italy
giulia.fracastoro@polito.it

Enrico Magli
Politecnico di Torino
Torino, Italy
enrico.magli@polito.it

ABSTRACT

Image denoising is a fundamental task in signal processing. Recent works have shown that data-driven approaches employing convolutional neural networks can outperform classical model-based techniques. These methods can capture highly complex image priors without the need to handcraft them. However, since these methods are based on convolutional operations, they are only capable of exploiting local similarities without taking into account non-local self-similarities, which have been highly successful in model-based methods. In order to exploit both local and non-local similarities, we introduce a graph-convolutional neural network specifically designed for image denoising. The graph-convolutional layers allow to dynamically construct neighborhoods in the feature space to detect spatially-distant pixels which show latent correlations in the the hidden layers.

1 INTRODUCTION

Recovering a clean image from a noisy observation is a crucial problem in signal processing. In order to address this problem, it is necessary to exploit prior knowledge about the structure of natural images. In the past, the literature on this topic has mainly focused on developing hand-crafted image priors that help to regularize this inverse problem. Popular methods that follow this approach include sparse representations (Elad & Aharon, 2006), total variation methods (Rudin et al., 1992; Pang & Cheung, 2017), and methods based on non-local self-similarities such as BM3D (Dabov et al., 2007), or WNNM (Gu et al., 2014), which are among the most successful ones. However, recent work (Zhang et al., 2017; Lefkimmiatis, 2018; Mao et al., 2016; Lehtinen et al., 2018) has shown that a data-driven approach, which employs convolutional neural networks, can outperform classical model-based techniques by capturing more complex and powerful image priors. Since these methods are based on convolutional operations, their main limitation is the local nature of the extracted features. Therefore, methods based on CNNs can only exploit local similarities and they are unable to capture non-local self-similar patterns that were proven to be highly successful in model-based methods.

In this work, we propose to overcome this limitation by employing a graph-convolutional neural network. Graph convolution is a generalization of the traditional convolution operation to process data with an irregular structure (Defferrard et al., 2016; Simonovsky & Komodakis, 2017; Kipf & Welling, 2016). In particular, we employ graph-convolutional layers in order to define in a more flexible way the neighborhood of each pixel. Using this approach, we can extract features that depend not only on spatially-adjacent pixels, but also on spatially-distant pixels which show nevertheless latent correlations by being close in the latent space. Notice that the proposed approach defines non-locality in a different manner with respect to the non-local neural network proposed in Wang et al. (2018) for video classification tasks, where the response at a position is computed by a weighted average of the features at all positions. Instead, in this paper we introduce a graph structure

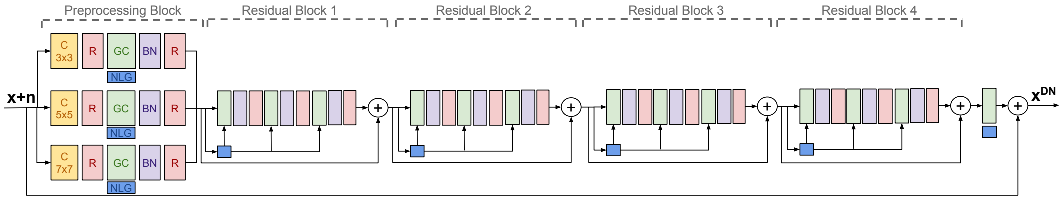

Figure 1: GraphCNN denoiser. Colored blocks are defined in the leftmost part. C: 2D convolution, R: leaky ReLU, GC: graph convolution (as in Fig.2), NLG: non-local graph construction (nearest neighbors in feature space), BN: batch normalization.

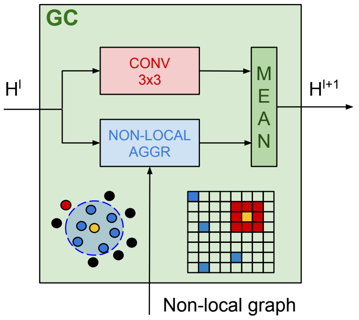

Figure 2: Graph-convolutional layer. Non-local aggregation block implements Eq. (1).

in order to select for each pixel only the most significant spatially-distant pixels, i.e. the closest in the feature space. This allows us to have a receptive field that can be dynamically adapted to the image characteristics. The obtained results show that the proposed graph-convolutional architecture outperforms traditional convolutional neural network for the denoising task.

2 Proposed method

In this section, we present the proposed architecture, called GraphCNN, to perform image denoising. An overview of the network is shown in Fig. 1. At a high-level, the network is composed of a sequence of graph-convolutional layers followed by batch normalization and leaky ReLU nonlinearities. The first part of the network is composed of a preprocessing block with parallel branches, reminiscent of similar constructions in Szegedy et al. (2015) and Divakar & Babu (2017). The goal of this first block is to extract features at multiple scales, by performing a classic convolution with three different filter sizes (3×3, 5×5, 7×7) followed by a graph convolution operation and finally concatenating the resulting features. The network also has several residual connections: notably, the input-output residual has been shown to be very effective for the denoising task Zhang et al. (2017) because it makes the network estimate noise features by progressively removing the clean image. The connection between the input and output of each residual block also improves gradient backpropagation.

2.1 Graph-convolutional layer

The graph-convolutional layer is at the core of the proposed model. A schematic representation is shown in Fig. 2. This layer extends the classical convolutional layer by aggregating the hidden-layer feature vectors of spatially-adjacent pixels as well as the hidden-layer feature vectors of spatially-distant pixels that are similar (nearest neighbors) in the feature space. The final output of the graph-convolutional layer is then an average between the local and non-local contribution. The local features are aggregated using a classic 3×3 convolution. Instead, the non-local features are aggregated using the edge-conditioned graph convolution as defined in Simonovsky & Komodakis (2017). Using this definition, the graph convolution operation performs weighted aggregations over a neighborhood, where the weights used for the aggregation depend on the edge labels of the graph. In particular, we define the edge label as the difference between the features of the two nodes of the edge. The weights of the local aggregation are defined by a fully-connected network $F^l : \mathbb{R}^{d^l} \rightarrow \mathbb{R}^{d^{l+1} \times d^l}$, which takes as input the edge labels and outputs the corresponding weight

Table 1: Set12 PSNR (dB)

	1	2	3	4	5	6	7	8	9	10	11	12	Avg
						$\sigma = 15$							
BM3D	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.372
WNNM	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	**33.60**	32.27	32.11	32.17	32.696
OGLR	31.36	34.88	32.31	30.70	31.26	30.46	30.87	33.97	32.54	31.58	31.59	31.71	31.936
DnCNN-S	**32.61**	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	**32.46**	32.47	32.859
GraphCNN	**32.61**	**35.16**	**33.34**	**32.49**	**33.34**	**31.89**	**31.92**	**34.64**	32.99	**32.49**	32.45	**32.48**	**32.985**
						$\sigma = 25$							
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.969
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	**31.24**	30.03	29.76	29.82	30.257
OGLR	29.11	32.65	30.02	28.29	29.16	28.10	28.76	31.95	30.35	29.59	29.47	29.49	29.744
DnCNN-S	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	**30.12**	30.436
GraphCNN	**30.19**	**33.38**	**30.92**	**29.86**	**30.58**	**29.32**	**29.51**	**32.54**	30.52	**30.28**	**30.13**	30.11	**30.614**
						$\sigma = 50$							
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.722
WNNM	26.45	**30.33**	26.95	25.44	26.32	25.42	26.14	29.25	**27.79**	26.97	26.94	26.64	27.052
OGLR	25.98	29.19	26.26	24.75	25.80	25.05	25.80	28.80	27.04	26.53	26.69	26.34	26.520
DnCNN-S	**27.03**	30.00	27.32	25.70	26.78	25.87	**26.48**	**29.39**	26.22	**27.20**	**27.24**	**26.90**	27.178
GraphCNN	27.02	**30.33**	**27.46**	**26.07**	**26.92**	**25.93**	26.41	29.38	26.93	27.10	27.19	26.85	**27.299**

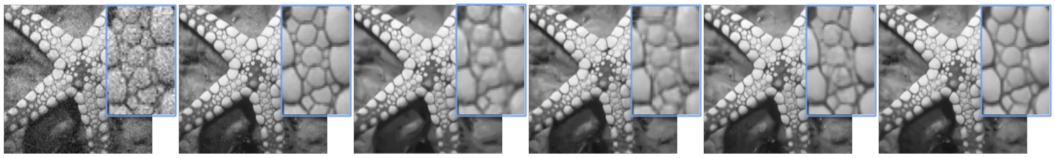

Figure 3: Denoising results for *starfish*. Left to right: noisy ($\sigma = 25$), original, OGLR (28.29 dB), BM3D (28.56 dB), DnCNN-S (29.41 dB), GraphCNN (**29.86 dB**).

matrix $\mathbf{\Theta}^{l,ji} = F^l_{\mathbf{w}^l}\left(\mathcal{L}(i,j)\right) \in \mathbb{R}^{d^{l+1} \times d^l}$. Then, for each node i of the graph we can define the convolution operation as follows

$$\mathbf{H}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}_i^l} \frac{F^l_{\mathbf{w}^l}\left(\mathbf{H}_j^l - \mathbf{H}_i^l\right)\mathbf{H}_j^l}{|\mathcal{N}_i^l|} + \mathbf{W}^l\mathbf{H}_i^l + \mathbf{b}^l\right) = \sigma\left(\underbrace{\sum_{j \in \mathcal{N}_i^l} \frac{\mathbf{\Theta}^{l,ji}\mathbf{H}_j^l}{|\mathcal{N}_i^l|}}_{\text{neighborhood}} + \underbrace{\mathbf{W}^l\mathbf{H}_i^l}_{\text{node}} + \mathbf{b}^l\right), \quad (1)$$

where $\mathbf{H}_i^l$ and $\mathcal{N}_i^l$ are respectively the feature vector and the neighborhood of the i-th node at the l-th layer, $\mathbf{w}^l$ are the weights parameterizing network F^l, $\mathbf{W}^l \in \mathbb{R}^{d^{l+1} \times d^l}$ is a linear transformation of the node itself, $\mathbf{b}^l$ a bias, and σ a non-linearity.

The non-local pixels are chosen as the k-nearest-neighbor feature vectors in terms of Euclidean distance with respect to the feature vector of the current pixel within a search window of predefined size. Notice that the non-local selection is performed only at some hidden layers (as shown by the NLG block in Fig. 1), with the two 3-layer residual blocks sharing the same non-local graph. This helps reducing complexity without compromising performance.

The role of the non-local graph in such residual architecture, whose goal is to successively remove the clean image from the noise features, is to identify the latent correlations in the feature space which are due to the residual image content rather than the uncorrelated noise.

We now analyze more in detail the role of the graph-convolution operation described in (1). Such definition provides two main contributions to the effectiveness of the algorithm. The key of this operation is the function F, which takes as input the difference between the feature vector of the current pixel and the feature vector of a non-local neighboring pixel and outputs the weight matrix used to transform the non-local feature vector before the aggregation. First, (1) can be called "convolution" because this function provides meaningful weight sharing under suitable stability assumptions: for a similar input difference, the output weight matrix should be similar. Second, differently from classical convolution this function enables a data-dependent aggregation because the weights depend directly on the relationships among feature vectors.

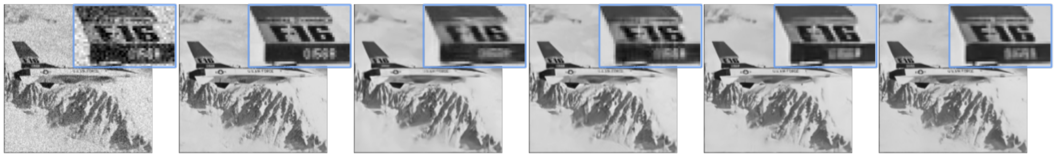

Figure 4: Denoising results for *airplane*. Left to right: noisy ($\sigma = 25$), original, OGLR (28.10 dB), BM3D (28.42 dB), DnCNN-S (29.13 dB), GraphCNN (**29.32 dB**).

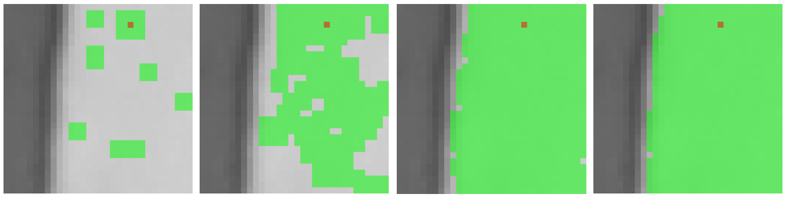

Figure 5: Receptive field (green) of a single pixel (red) for graph-convolutional layers 1-4 (ignoring the 5×5 and 7×7 multiscale branches for ease of representation).

3 Results

In this section we perform an experimental evaluation of the proposed method, comparing the proposed method with several state-of-the-art denoising methods. We consider two model-based methods that exploit non-local priors (i.e., BM3D (Dabov et al., 2007) and WNNM (Gu et al., 2014)), a graph-based variational method (i.e., OGLR (Pang & Cheung, 2017)) and the state-of-the-art CNN model for image denoising (i.e., DnCNN (Zhang et al., 2017)).

3.1 Experimental settings

In the following experiments, we consider grayscale images. For training, we use the 432 training images of the BSD500 dataset (Arbelaez et al., 2011). Instead, for testing we use a set of 12 widely used images (i.e., Cameraman, House, Peppers, Starfish, Monarch, Airplane, Parrot, Lena, Barbara, Boat, Man, Couple). We train the network using a fixed noise standard deviation, considering three different noise levels $\sigma = 15, 25, 50$. We subdivide the images into patches of size 32×32 and train the network on 200k patches for 30 epochs. The non-local graph selects the 8 nearest neighboring pixels in terms of Euclidean distance between the hidden feature vectors, excluding the spatially adjacent pixels. The number of hidden features is 66 for all layers, except for the ones in the branches of the preprocessing block, for which is 22.

3.2 Quantitative and qualitative results

Table 1 shows the PSNR results on the 12 images of the test set. We can see that the proposed architecture outperforms the competing methods on most of the images and it provides the best average scores. In order to highlight the importance of the non-local filters, Table 2 compares the proposed method with a network having the same architecture of the proposed model but employing only local neighbors instead of local and 8 non-local. The results show that the nonlocal model is indeed key to achieving the best possible performance. Notice that the 0-NN GraphCNN and the the DnCNN-S show similar performance, which suggests that the gain provided by the 8-NN Graph CNN is due to the non-local contribution. In addition to these quantitative results, we also show a qualitative comparison of the denoising methods in Figs. 3 and 4. We can clearly see that the proposed method provides the best visual quality, recovering finer details and producing fewer artifacts. Lastly, we show in Fig. 5 the receptive field of a pixel for the first four graph-convolutional

Table 2: Set12 average PSNR (dB) without non-local NN

	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
DnCNN-S	32.859	30.436	27.178
GraphCNN (0-NN)	32.858	30.411	27.100
GraphCNN (8-NN)	32.985	30.614	27.299

layers. It is interesting to note that the receptive field is adapted to the image characteristics, covering only a homogeneous region of the image.

REFERENCES

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16 (8):2080–2095, 2007.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

Nithish Divakar and R. Venkatesh Babu. Image denoising via CNNs: an adversarial approach. In *New Trends in Image Restoration and Enhancement, CVPR*, 2017.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec 2006.

Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2862–2869, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Stamatios Lefkimmiatis. Universal denoising networks: a novel CNN architecture for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3204–3213, 2018.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: learning image restoration without clean data. In *International Conference on Machine Learning (ICML)*, 2018.

Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2802–2810. Curran Associates, Inc., 2016.

Jiahao Pang and Gene Cheung. Graph Laplacian regularization for image denoising: analysis in the continuous domain. *IEEE Transactions on Image Processing*, 26(4):1770–1785, 2017.

Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29–38, July 2017.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, June 2015.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

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