NIR image colorization with graph-convolutional neural networks

NIR image colorization with graph-convolutional neural networks

Diego Valsesia, Giulia Fracastoro, Enrico Magli
Politecnico di Torino, Italy

***Abstract*—Colorization of near-infrared (NIR) images is a challenging problem due to the different material properties at the infared wavelenghts, thus reducing the correlation with visible images. In this paper, we study how graph-convolutional neural networks allow exploiting a more powerful inductive bias than standard CNNs, in the form of non-local self-similiarity. Its impact is evaluated by showing how training with mean squared error only as loss leads to poor results with a standard CNN, while the graph-convolutional network produces significantly sharper and more realistic colorizations.**

***Index Terms*—NIR colorization, graph neural network**

I. Introduction

Imaging in the near infrared (NIR) spectrum has found several applications in video surveillance, and remote sensing. The main feature offered by the NIR band is a tighter dependence of the surface reflectance to the material, as many dyes and pigments are transparent at infrared wavelengths. This is a valuable property when a system is tasked with segmenting a scene [1] or detecting objects based on material properties. However, it also makes the scene more challenging to understand by humans due to the lack of color references. Colorization, i.e., the process of producing a RGB image from a monochrome input, would allow to make NIR scenes more interpretable. However, contrary to visible image colorization [2], where the task is to recover the chrominance signal from the luminance component only, NIR colorization is a more challenging problem. NIR image colorization lacks the low-level clues available in the luminance channel of visible images due to the different material properties outside the visible spectrum. The colorization task must therefore learn higher-level semantic information in order to successfully deduce a plausible color.

The colorization problem, including NIR colorization, is typically tackled by exploiting generative adversarial networks (GANs) [3]–[6] in addition to pixel-wise losses. In fact, pixel-wise losses such as mean-squared error (MSE) or angular error (AE), or even pseudo-perceptual metrics such as SSIM, are not sufficient to regularize the problem so that the colorized image sticks to the manifold of realistic RGB images [7], which is needed to produce convincing results.

In this paper, however, we focus on a complementary issue to the choice of loss function. We argue that the network architecture itself can provide a valuable inductive bias, that, if properly designed, provides an extra source of regularization. We argue that the inductive bias provided by classic convolutional neural network (CNN) architectures is insufficient to fully address the NIR colorization problem. In fact, the highly localized receptive field of CNNs allows them to focus more on local statistics. We propose to use graph-convolutional neural networks to exploit non-local self-similarities and learn more global features. In order to show that the inductive bias due to the graph-convolution operation is effective on this task, we focus on training with a pixel-wise MSE only. While the classic CNN dramatically fails to provide meaningful colorizations, the graph-convolutional neural network is able to generate sensible results. This is a remarkable result due to the aforementioned limitations of the MSE loss, and shows that graph-convolution can provide a better backbone network, paving the way to its use in future works including adversarial training.

978-1-7281-8068-7/20/$31.00 ©2020 IEEE

II. Background on graph-convolutional neural networks

In the last years, graph-convolutional neural networks have emerged as the state-of-the-art approach when dealing with data defined on irregular domains. The main building block of such networks is the graph-convolutional layer. Numerous definitions of graph convolution have been proposed. A first class of approaches defines the graph convolution operation in the spectral domain through the graph Fourier transform [8]–[10]. In order to reduce the computational complexity of this operation, fast polynomial approximations have been proposed [9]. One of the most famous graph neural network architectures, i.e, the Graph Convolutional Network (GCN) presented in [10], employs a first order polynomial approximation. The main drawback of such spectral approaches is that the graph structure is supposed to be fixed and it cannot handle the cases where the graph structure varies. In order to overcome this issue, a second class of approaches was proposed. In this case, the graph convolution operator is defined in the spatial domain: the convolution is performed as a weighted local aggregation over the neighboring nodes [11]–[18]. Thanks to the fact that the convolution is defined at a neighborhood level, the operation remains well defined even if the graph structure changes. Various techniques for defining the weights employed in the neighborhood aggregation have been proposed. Several methods use learnable weights whose values are the same for all the nodes of the graph [13], [16]. Instead, in other cases the weights depend on the input features of the node. For example, [14] computes the aggregation weights using a

Gaussian kernel, instead [15] employs a soft-max function with learnable parameters. A more general approach, called Edge Conditioned Convolution (ECC), is proposed by [11], where the function that computes the aggregation weights is defined using a multilayer perceptron (MLP). This makes the function highly general since it does not impose a predefined structure, but the approximation capability of the MLP allows to learn the optimal function for a given specific task. A more efficient version of the ECC, called Lightweight ECC, is proposed in [19], where some approximations are introduced in order to reduce the number of parameters and the memory usage. In [19], the lightweight ECC has been effectively applied in the context of image denoising. In this case, the use of graph-convolutional layers allows to exploit both local and non-local similarities that are present in the image by creating a graph that connects pixels whose feature representations are similar to each other, regardless of their spatial distance. This approach outperforms standard CNNs, which can exploit only local information.

III. Proposed method

This section presents NIR-GNN, the proposed graph-convolutional architecture for NIR image colorization. An overview is shown in Fig. 1. At a first glance, we can observe that the proposed architecture has a global input-output residual connection whereby the network learns to estimate the difference between each RGB channel of the image and the monochromatic NIR image rather than directly produce the RGB image.

The first block of the proposed network is a preprocessing stage with three parallel branches that operate on multiple scales. The multiscale features are extracted by a sequence of three 2D convolutional layers with filters of size 3×3, 5×5, and 7×7, depending on the branch. Then, after a final graph-convolutional layer, the features are concatenated.

The remainder of the network is composed by an HPF block and two LPF blocks, named after an analogy with highpass and lowpass graph filters of an unrolled proximal gradient descent optimization method. We refer the reader to [19] for more theoretical details on such analogy. All these blocks are composed of an initial 3×3 2D convolutional layer followed by three graph-convolutional layers. A graph is constructed as a k-nearest neighbor graph by evaluating the Euclidean distance between the feature vectors of each pixel. To reduce computational complexity, all the graph-convolutional layers in a block share the same graph, which is constructed from the output of the initial 3×3 convolutional layer. The graph is then updated at the next block, creating a dynamic construction that is better adapted to the evolving feature space. All layers are interleaved by Batch Normalization operations and leaky ReLU nonlinearities. Moreover, the LPF blocks have an input-output skip connection to promote backpropagation, as in ResNet architectures. Finally, a last graph-convolutional layer projects the features back to the RGB space.

The main building block of the proposed network is the graph-convolutional layer. Each graph-convolutional layer has two inputs: a matrix $\mathbf{H}^{l+1} \in \mathbb{R}^{F^l \times N}$ representing a feature vector with dimension F^l for each of the N pixels of the patch, and a graph describing the connections between pixels. In order to exploit both the local and non-local similarities of the image, for each pixel we perform two different types of aggregation: a local aggregation of the pixels that are spatially close, and a non-local aggregation of pixels that are spatially distant but have similar feature representations. A standard 3×3 2D convolution processes the local neighborhood, while the lightweight ECC [19] is employed to aggregate the non-local neighbors, which are defined by the k-nearest neighbor graph. Then, the output feature vectors $\mathbf{H}^{l+1} \in \mathbb{R}^{F^{l+1} \times N}$ at layer l are computed summing these two contributions as follows:

$$\begin{aligned}\mathbf{H}_i^{l+1} &= \frac{\mathbf{H}_i^{l+1,\mathrm{L}} + \mathbf{H}_i^{l+1,\mathrm{NL}}}{2} + \mathbf{b}^l \\ &= \frac{1}{2}\mathbf{H}_i^{l+1,\mathrm{L}} + \sum_{j \in \mathcal{N}_i^l} \frac{\gamma^{l,j \to i} \sum_{t=1}^{r} \omega_t^{j \to i} \boldsymbol{\phi}_t^{j \to i} \boldsymbol{\psi}_t^{j \to i^T} \mathbf{H}_j^l}{2|\mathcal{N}_i^l|} + \mathbf{b}^l,\end{aligned} \tag{1}$$

where $\mathbf{H}_i^l$ is the input feature vector at pixel i, $\mathcal{N}_i^l$ is the set of its non-local neighbors, $\mathbf{H}_i^{l+1,\mathrm{L}}$ is the output of the 3×3 local convolution for pixel i, $\mathbf{H}_i^{l+1,\mathrm{NL}}$ is the output of the lightweight ECC for pixel i, and $\mathbf{b}^l \in \mathbb{R}^{F^l}$ is the bias. The weights employed in the non-local aggregation, i.e., vectors $\boldsymbol{\phi}_t^{j \to i} \in \mathbb{R}^{F^{l+1}}$, $\boldsymbol{\psi}_t^{j \to i} \in \mathbb{R}^{F^l}$ and $\omega_t^{j \to i} \in \mathbb{R}$, are computed as functions of the difference between the input feature vectors of point i and point j, i.e., $\boldsymbol{\phi}_t^{j \to i}, \boldsymbol{\psi}_t^{j \to i}, \omega_t^{j \to i} = \mathcal{F}\left(\mathbf{H}_i^l - \mathbf{H}_j^l\right)$. This function is defined as a MLP with two layers. Due to overparameterization issues, the last layer of the MLP is approximated employing a stack of circulant matrices as done in [19]. In (1), the non-local aggregation weight matrix is approximated as $\sum_{t=1}^{r} \omega_t^{j \to i} \boldsymbol{\phi}_t^{j \to i} \boldsymbol{\psi}_t^{j \to i^T}$, where r is a hyperparameter setting the maximum rank; this is done to reduce the number of parameters and memory requirements of the aggregation operation. We refer the reader to [19] for additional details on these approximations. The scalar multiplier $\gamma^{l,j \to i} \in \mathbb{R}$ is an edge attention term which depends on the Euclidean distance between feature vectors of neighboring nodes:

$$\gamma^{l,j \to i} = \exp\left(-\|\mathbf{H}_i^l - \mathbf{H}_j^l\|_2^2 / \delta\right),$$

where δ is a decay hyperparameter.

IV. Experimental results

This section presents some experimental results on the dataset provided for the VCIP 2020 Grand Challenge on NIR Image Colorization. The focus of our experiments is to compare the images colorized by a CNN and a by the proposed graph-convolutional neural network when trained with a MSE loss. This is done to ensure that the experiment evaluates the merits of the network design, rather than introducing confounding factors like adversarial training. Fairness is ensured by using a roughly similar number of parameters and same overall architecture, but without the non-local graph-convolutional contribution in the standard CNN.

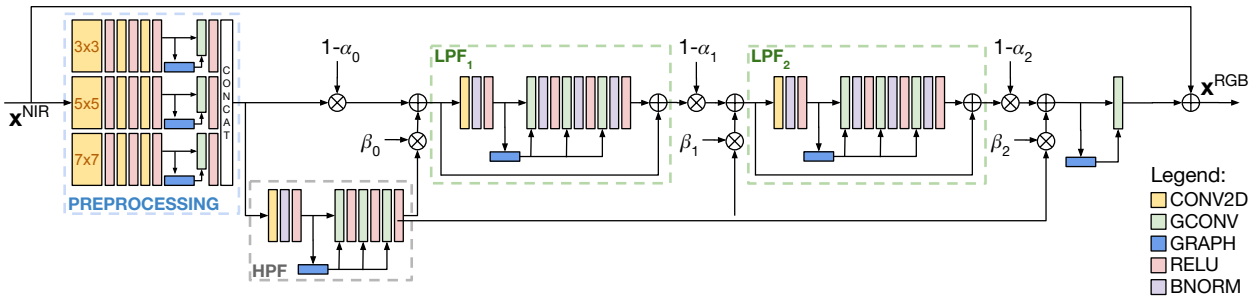

Figure 1. NIR-GNN: proposed graph-convolutional architecture.

Figure 2. Colorized *validation_0004*. Left to right: NIR image, CNN, NIR-GNN, RGB ground truth.

Figure 3. Colorized *validation_0013*. Left to right: NIR image, CNN, NIR-GNN, RGB ground truth.

Figure 4. Colorized *validation_0016*. Left to right: NIR image, CNN, NIR-GNN, RGB ground truth.

Table I
OBJECTIVE QUALITY METRICS. VALIDATION SET.

	PSNR	**SSIM**	**AE**
CNN	16.17 dB	0.479	0.32
NIR-GNN	**16.27 dB**	**0.514**	0.32

We trained only on the available paired NIR−RGB image dataset, without using any external data or unpaired images. Both networks were trained for 300000 iterations with a learning rate exponentially decayed from 10^{-4} to 10^{-5}, each processing a batch of 10 patches of size 42×42. The graph-convolutional network used $F = 99$ features, $r = 11$ rank, $\delta = 10$ and $k = 8$ non-local nearest neighbors.

Figs.2,3, and 4 show a validation image, as colorized by the standard CNN and by NIR-GNN. As expected, the MSE loss is insufficient to provide meaningful colorization results using the standard CNN. The image is quite blurry and the colors have weak spatial consistency, revealing color patterns that do not exist in the ground truth. The images generated by NIR-GNN are significantly sharper, colors more accurately follow the scene content, also showing good variability, consistent with the objects in the scene. Notice how both images lack vibrant colors: this can be attributed to the limitations of the MSE loss.

Finally, Table I summarizes the results on objective quality metrics, averaged over the entire validation set. Notice how the metrics are not able to fully capture the visual quality improvements.

V. CONCLUSIONS

We presented a neural network based on graph-convolutional layers to tackle the NIR image colorization problem. Our objective was to show how this constitutes a better backbone than traditional CNNs since the graph convolution operation enables a strong inductive bias in the form of the exploitation of non-local self-similar features. This enhanced prior allows the colorized images to be more realistic, with sharper details and spatially-consistent color distributions, even when a simple MSE is used. Future works should consider integrating the proposed method with adversarial losses to further improve the results.

REFERENCES

[1] Neda Salamati, Diane Larlus, Gabriela Csurka, and Sabine Süsstrunk, "Semantic image segmentation using visible and near-infrared channels," in *European Conference on Computer Vision*. Springer, 2012, pp. 461–471.

[2] Saeed Anwar, Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, and Abdul Wahab Muzaffar, "Image colorization: A survey and dataset," *arXiv preprint arXiv:2008.10774*, 2020.

[3] Patricia L Suárez, Angel D Sappa, and Boris X Vintimilla, "Infrared image colorization based on a triplet dcgan architecture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 18–23.

[4] Matthias Limmer and Hendrik PA Lensch, "Infrared colorization using deep convolutional neural networks," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 61–68.

[5] Amanda Berg, Jorgen Ahlberg, and Michael Felsberg, "Generating visible spectrum images from thermal infrared," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1143–1152.

[6] Seungjoo Yoo, Hyojin Bahng, Sunghyo Chung, Junsoo Lee, Jaehyuk Chang, and Jaegul Choo, "Coloring with limited data: Few-shot colorization via memory augmented networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11283–11292.

[7] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[8] Mikael Henaff, Joan Bruna, and Yann LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

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

[10] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[11] Martin Simonovsky and Nikos Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 29–38.

[12] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[13] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, "How powerful are graph neural networks?," in *International Conference on Learning Representations*, 2019.

[14] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5115–5124.

[15] Nitika Verma, Edmond Boyer, and Jakob Verbeek, "Feastnet: Feature-steered graph convolutions for 3d shape analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2598–2606.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[18] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković, "Principal neighbourhood aggregation for graph nets," *arXiv preprint arXiv:2004.05718*, 2020.

[19] Diego Valsesia, Giulia Fracastoro, and Enrico Magli, "Deep graph-convolutional image denoising," *IEEE Transactions on Image Processing*, vol. 29, pp. 8226–8237, 2020.