PNEN: PYRAMID NON-LOCAL ENHANCED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing neural networks proposed for low-level image processing tasks are usually implemented by stacking convolution layers with limited kernel size. Every convolution layer merely involves in context information from a small local neighborhood. More contextual features can be explored as more convolution layers are adopted. However it is difficult and costly to take full advantage of long-range dependencies. We employ non-local operation to build up connection between every pixel and all remain pixels. Moreover a novel *Pyramid Non-local Block* is devised to robustly estimate pairwise similarity coefficients between different scales of content patterns. Considering computation burden and memory consumption, we exploit embedding feature maps with coarser resolution to represent content patterns with larger spatial scale. Through elaborately combining the pyramid non-local blocks and dilated residual blocks, we set up a *Pyramid Non-local Enhanced Network* for edge-preserving image smoothing. It achieves state-of-the-art performance in imitating three classical image smoothing algorithms. Additionally, the pyramid non-local block can be directly incorporated into existing convolution neural networks for other image processing tasks. We integrate it into two state-of-the-art methods for image denoising and single image super-resolution respectively, achieving consistently improved performance.

1 INTRODUCTION

Recently impressive progress has been achieved in low-level computer vision tasks as the development of convolution neural networks (CNN), e.g. edge-preserving image smoothing (Li et al., 2016; Fan et al., 2017; Zhu et al., 2019), image denoising (Mao et al., 2016; Zhang et al., 2017; 2018) and image super-resolution (Tai et al., 2017a;b; Zhang et al., 2018). This paper focuses on devising a novel pyramid non-local block oriented for effectively and efficiently mining long-range dependencies in low-level image processing tasks.

Typical convolution layers operate on a small local neighborhood without considering non-local contextual information. One common practice for capturing long-range dependencies is to enlarge the receptive field by stacking large number of convolution layers (Kim et al., 2016a) or dilating convolution layers (Yu et al., 2017). However, it is difficult to deliver information between distant positions in such a manner (Hochreiter & Schmidhuber, 1997).

Discovering similar patterns (including small texture patches and large object parts) residing in natural images is valuable to low-level image processing (Buades et al., 2005; Dabov et al., 2007; Mairal et al., 2009). Dependencies on similar patterns can be used for recognizing out real region boundary in image smoothing and recovering contaminated or missing details in image restoration tasks. Inspired from this concept, a few literatures (Lefkimmiatis, 2017; Wang et al., 2018; Liu et al., 2018; Li et al., 2018a) introduce non-local operation into deep networks in form of self-similarity strategy. Wang et al. (2018) presents a non-local component in video classification, which is placed after high-level, sub-sampled feature maps. Huge computational cost and memory consumption of their non-local operations hinder its adaption to low-level computer vision tasks, where high-resolution feature maps are demanded to produce appealing pixel-wise outputs. NLRN (Liu et al., 2018) and NLEDN (Li et al., 2018a) apply non-local operation for image restoration and rain removal, respectively. One fatal drawback of this kind of methods is that the estimation of self-similarity is confined in a neighborhood of tens of pixels. Besides it is a very common phenomenon that the same type of patterns appear to own various spatial scales, which has not been taken into consideration in existing non-local operations.

To settle the above issues, we propose a pyramid structure, named *Pyramid Non-local Block* (PNB), to perform non-local operation with a cost-effective amount of computation load. It utilizes a query feature map with full resolution and a pyramid of reference feature maps to robustly estimate similarities between different scales of patterns. Accordingly, a pyramid of embedding feature maps are extracted to enhance the input feature map with the help of estimated similarity matrices. To ensure acceptable computation burden and memory consumption, resolutions of the reference and embedding feature maps are downscaled. Through intervening pyramid non-local blocks and dilated residual blocks (Yu et al., 2017), we set up *Pyramid Non-Local Enhanced Networks* for edge-preserving image smoothing. It achieves the state-of-the-art performance in imitating various classical image smoothing filters.

In addition, efficient computation allows our proposed pyramid non-local block to be incorporated into deep CNN-based methods for pixel-level image processing tasks. We demonstrate the effectiveness of PNB on two classical tasks, image denoising and single image super-resolution (SISR). Two state-of-the-art models, RDN (Zhang et al., 2018) and MemNet (Tai et al., 2017b), are adopted as the baseline models. The PNB acts as a critical component to exploit long-range dependencies. Performance improvements over baseline have been consistently obtained thanks to the adoption of pyramid non-local blocks.

2 RELATED WORK

Deep Learning Based Image Processing Edge-preserving image smoothing aims to preserve significant image structures while smoothing out trivial details. Because of the strong feature learning ability, deep neural networks attracts a lot of attention in this field. The pioneering work (Xu et al., 2015) employs a three-layer CNN to predict a gradient map which is subsequently used to guide the smoothing procedure. Recurrent network is adopted to efficiently propagate spatial contextual information across pixels (Liu et al., 2016). Joint filtering (Li et al., 2016) and edge map (Fan et al., 2017) are dedicated to using extra guidances for image smoothing. Meanwhile, deep neural networks have been proposed for image denoising and super-resolution (Kim et al., 2016a; Zhang et al., 2017). Residual connections (Lim et al., 2017) and dense connections (Tong et al., 2017; Zhang et al., 2018) are added to alleviate the vanishing-gradient problem. Features at different scales are fused to get efficacious information (Li et al., 2018b). Other representative methods include Huang et al. (2015); Shi et al. (2016); Tai et al. (2017a;b); Ahn et al. (2018); Hui et al. (2018). We do not elaborate due to limited space.

Non-local Context Information. Image non-local self-similarity has been widely exploited in many non-local methods for image restoration, such as BM3D (Dabov et al., 2007) and WNNM (Gu et al., 2014). Recently a few studies attempt to incorporate non-local operations into deep neural networks for capturing long-range dependencies. Wang et al. (2018) presented trainable non-local neural networks in video classification. However, the computation complexity of their non-local operation grows dramatically as size of the input feature map increases. Liu et al. (2018) and Li et al. (2018a) incorporated non-local module into RNN architecture and encoder-decoder architecture for image restoration and image de-raining, respectively. Nevertheless, the measurement of self-similarity is restricted within a small neighborhood.

Our method differs from existing models proposed for low-level image processing in two aspects. Firstly, our method can robustly measure similarities between different scales of content patterns as we adopt a pyramid structure for the non-local operation. Secondly, computation burden and memory consumption is greatly relieved because the spatial resolutions of reference features in the non-local operation are downscaled.

3 METHOD

Non-local correlations between patterns are paramount for edge-preserving image smoothing. Long-range contextual features are beneficial to suppress textures inside objects (object parts) while identifying out real region boundaries in image smoothing. We propose a deep pyramid non-local

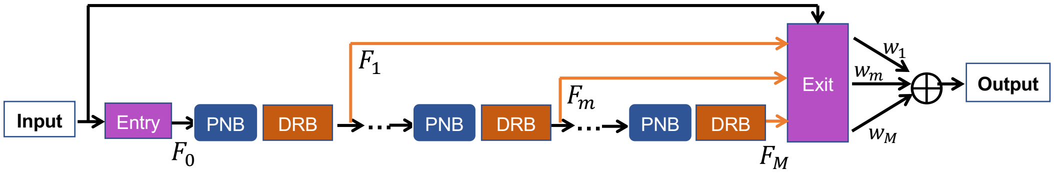


Figure 1: The overall architecture of our proposed pyramid non-local enhanced network (PNEN).

enhanced network (PNEN), which introduces pyramid non-local block (PNB) to explicitly mine long-range dependencies. The overall architecture is illustrated in Figure 1. The pyramid non-local block (PNB) is carefully designed to involve in correlation with distant pixels when inferring every pixel, as well as guaranteeing computation efficiency in terms of complexity and tolerant memory consumption. Besides, dilated residual blocks (DRB) (Yu et al., 2017) are introduced to enlarge receptive field for taking advantage of full structural and textural information in the image. In the following sections, each component of the proposed architecture will be elaborated with more details.

3.1 Entry and Exit Network

Define the input color image as $\mathbf{X}$ with size of $h \times w \times 3$. h and w represents the height and width of the input image respectively. Our proposed PNEN employs one convolution layer as the entry net to transform $\mathbf{X}$ into a feature map $\mathbf{F}_0$ which is a $h \times w \times c$ tensor. Formally, we have

$$\mathbf{F}_0 = \mathcal{F}_{entry}(\mathbf{X}, \mathbf{W}_{entry}), \tag{1}$$

where the $\mathcal{F}_{entry}(\cdot, \cdot)$ denotes the convolution operations in the entry net and $\mathbf{W}_{entry}$ represents related convolution parameters. Subsequently, M blocks, each of which is consisting of a pyramid non-local block and a dilated residual block, are stacked to induce deep features. We define the feature produced by the m-th block as $\mathbf{F}_m$. We have

$$\mathbf{F}_m = \mathcal{F}_{PNB}(\mathcal{F}_{DRB}(\mathbf{F}_{m-1}, \mathbf{W}^m_{DRB}), \mathbf{W}^m_{PNB}), \tag{2}$$

where $\mathcal{F}_{PNB}(\cdot, \cdot)$ and $\mathcal{F}_{DRB}(\cdot, \cdot)$ indicates the calculation procedure inside the pyramid non-local block and dilated residual block which will be elaborated in Section 3.2 and 3.3 respectively. $\mathbf{W}^m_{DRB}$ and $\mathbf{W}^m_{PNB}$ represent their parameters correspondingly. Inspired by MemNet (Tai et al., 2017b), features generated by all blocks $\{\mathbf{F}_m | m = 1, \cdots, M\}$ are accumulated to generate residual images using the exit network. The residual image produced with $\mathbf{F}_m$ is defined as

$$\mathbf{R}_m = \mathcal{F}_{exit}(\mathbf{F}_m, \mathbf{W}^m_{exit}), \tag{3}$$

where $\mathcal{F}_{exit}(\cdot, \cdot)$ denotes the convolution operations in the exit network and $\mathbf{W}^m_{exit}$ represents the related parameters. The exit network contains three convolutional layers. Suppose $\mathbf{Y}_m = \mathbf{X} + \mathbf{R}_m$. The final reconstructed images is computed by

$$\mathbf{Y} = \sum_1^M w_m \cdot \mathbf{Y}_m, \tag{4}$$

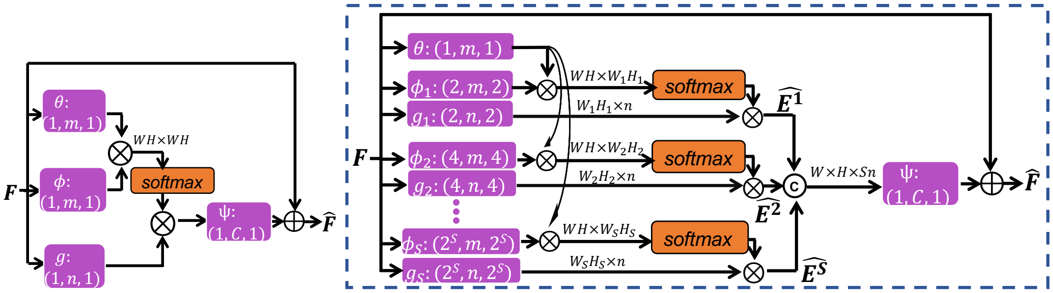
where $\{w_m | m = 1, \cdots, M\}$ are trainable weights. During the training stage, supervisions are imposed to intermediate predictions $\mathbf{Y}_m$-s and final output $\mathbf{Y}$. The loss function is defined as the mean squared error towards groundtruth images $\mathbf{G}$:

$$\mathcal{L} = \frac{1}{hwc}(\|\mathbf{G} - \mathbf{Y}\|^2 + \sum_{m=1}^M \|\mathbf{G} - \mathbf{Y}_m\|^2). \tag{5}$$

3.2 Pyramid Non-local Block (PNB)

Let $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$ denotes the input feature activation map. Here h, w and c represents the height, width and channel, respectively. A general formulation of non-local operation (Wang et al., 2018) can be defined as

$$\hat{\mathbf{F}} = \mathcal{T}(\frac{1}{\mathcal{D}(\mathbf{F})}\mathcal{M}(\mathbf{F})\mathcal{G}(\mathbf{F})) + \mathbf{F}, \tag{6}$$



(a) Non-local Block

(b) Pyramid Non-local Block

Figure 2: Architecture of prior non-local block (Wang et al., 2018) and our pyramid non-local block (PNB). In PNB, ϕ_k and g_k are implemented with convolutional layers of different strides, while θ is shared. The kernel size *k*, number of filters *f* and stride *s* are indicated as (*k*, *f*, *s*) for each convolution layer.

where $\hat{\mathbf{F}}$ is the enhanced feature representation. $\mathcal{M}(\mathbf{F}) \in \mathbb{R}^{hw \times hw}$ is the self-similarity matrix, where each element $\mathcal{M}(\mathbf{F})_{i,j}$ indicates the similarity between pixel i and j. $\mathcal{G}(\mathbf{X}) \in \mathbb{R}^{hw \times n}$ gives rise to a n-dimensional pixel-wise embedding. $\mathcal{D}(\mathbf{X})$ is a diagonal matrix for normalization purpose. $\mathcal{T}(\cdot)$ is the transformation function which converts the embedded features back into the original space of input feature. In this way, the feature representation is non-locally enhanced through considering all positions of the feature map. One instantiation (Wang et al., 2018) can be constructed by taking the linear embedded Gaussian kernel as the distance metric to compute correlation matrix $\mathcal{M}$, and linear function to compute $\mathcal{G}$:

$$\mathcal{M}(\mathbf{F}) = \exp(\mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_\theta)\mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_\phi)^{\mathrm{T}}), \tag{7}$$

$$\mathcal{G}(\mathbf{F}) = \mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_g). \tag{8}$$

The embedding function $\mathcal{F}_{emb}(\mathbf{F}, \mathbf{W})$ can be implemented by first applying convolutional operation of parameter $\mathbf{W}$ on $\mathbf{F}$, and then transforming the result into a 2-dimensional tensor in which each column represents one embedding channel. When calculating $\mathcal{M}(\mathbf{F})$, a query and a reference feature with same size of $hw \times m$ are generated using convolution kernel $\mathbf{W}_\theta$ and $\mathbf{W}_\phi$ respectively. The output dimension of $\mathbf{W}_g$ is denoted as n. The diagonal elements of $\mathcal{D}(\mathbf{F})$ are obtained through calculating column summation of $\mathcal{M}(\mathbf{F})$. $\mathcal{T}(\cdot)$ is also implemented with a convolution operation of parameter $\mathbf{W}_\psi$ to convert the embedding feature back into the original c-dimensional space. All convolutions use kernel size of 1×1. An example non-local block is illustrated in Figure 2(a). The computation complexity and memory occupation of the correlation matrix increases exponentially as the number of pixels grows. For sake of reducing computation burden, previous works (Liu et al., 2018; Li et al., 2018a) utilize a small neighborhood to restrict the range of non-local operation. In comparison, we propose a novel pyramid non-local block to effectively soften the computation demand while achieving appealing performance in low-level image processing tasks.

At first, we produce one query feature $\mathbf{E}_\theta = \mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_\theta)$. The spatial kernel size and stride of $\mathbf{W}_\theta$ is 1×1 and 1 respectively. Then, multi-scale reference features and embedding features are generated with convolutional layers using different kernel sizes and strides. Suppose there are totally S scales. The computation procedure of the non-local operation in the s-th scale is as follows:

$$\mathbf{E}_\theta = \mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_\theta), \quad \mathbf{E}_\Phi^s = \mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_\Phi^s), \quad \mathbf{E}_g^s = \mathcal{F}_{emb}(\mathbf{F}, \mathbf{W}_g^s), \tag{9}$$

$$\hat{\mathbf{E}}^s = \frac{1}{\mathbf{D}^s}\exp\{\mathbf{E}_\theta(\mathbf{E}_\Phi^s)^{\mathrm{T}}\}\mathbf{E}_g^s, \tag{10}$$

where $\{\mathbf{E}_\Phi^s | s = 1, \cdots, S\}$ are query features for calculating similarity matrices with respect to $\mathbf{E}_\theta$ and $\{\mathbf{E}_g^s | s = 1, \cdots, S\}$ are embedding features in all scales. The stride of convolutional layers in the s-th scale is set to 2^s. This implies that the number of rows in $\mathbf{E}_\Phi^s$ and $\mathbf{E}_g^s$ is reduced to $hw/4^s$, which greatly reduces the amount of computation for estimating the self-similarity matrix. We adopt larger convolutional kernels to extract reference feature representations of larger content patterns. The pyramidal design is motivated by two reasons: 1) The computation and memory requirement of non-local operations can be controlled within an acceptable load. 2) The robustness of self-similarity estimation is enhanced since the self-similarity is measured at various resolutions.

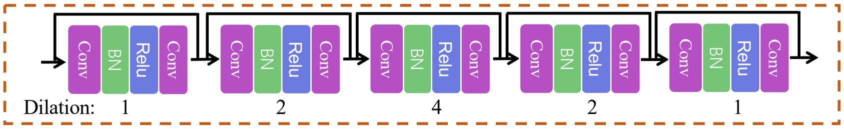


Figure 3: Architecture of dilated residual block (DRB). Our proposed DRB contains five Conv-BN-ReLU-Conv groups, which have dilation factor 1, 2, 4, 2, 1, respectively.

After performing non-local operation in all scales, enhanced embedding features $\{\hat{\mathbf{E}}^s | s = 1, \cdots, S\}$ are concatenated together, followed by one 1×1 convolution layer to generate residual values to $\mathbf{F}$. Formally, the final output of pyramid non-local block can be achieved by

$$\hat{\mathbf{F}} = \mathcal{F}_\Psi(\{\hat{\mathbf{E}}^1, \cdots, \hat{\mathbf{E}}^S\}, \mathbf{W}_\psi) + \mathbf{F}. \tag{11}$$

We summarize Eq. (9), (10) and (11) with the function $\mathcal{F}_{PNB}(\cdot, \cdot)$, as mentioned in Section 3.1. One characteristic of pyramid non-local block is the flexibility of balancing accuracy and computation resources through adjusting kernel sizes and strides in different scales. An illustration of the pyramid non-local block is shown in Figure 2(b). We set $m = 64, n = 32$ and $S = 3$ in practice.

3.3 Dilated Residual Block (DRB)

In pixel-wise image processing tasks, high-resolution feature maps are favorable for reconstructing complicate textural details while large receptive field benefits the extraction of features which can grab high-level contextual information. Considering the above issues, we employ dilated convolution (Yu et al., 2017) to rapidly increase the receptive fields without sacrificing spatial resolutions of learned features in our proposed model. As shown in Figure 3, we devise a dilated residual block with 5 cascaded residual modules which is set up with dilated convolutions. The calculation procedure of the dilated residual block is indicated with the function $\mathcal{F}_{DRB}(\cdot, \cdot)$, as mentioned in Section 3.1.

As illustrated in Figure 1, our model is built up through intervening PNB-s and DRB-s. Every group of consecutive PNB and DRB contributes a feature map for the final prediction.

3.4 Discussion

The benefits of our proposed pyramid non-local block are three folds: First, the pyramidal strategy adopts convolutions with various kernel sizes to generate embedding features for self-similarity estimation. This improves the robustness of estimating correlation across pattern scales.

Second, existing deep models equipped with non-local modules are implemented via connecting all pairs of pixels in the feature map Wang et al. (2018) or limiting the nonlocal dependencies within a constant neighborhood size Liu et al. (2018); Li et al. (2018a). The former kind of methods merely plug non-local modules after high-level feature maps with small resolution, because the computational complexity and memory usage will grow exponentially as the number of pixels in the feature map increases. The later kind of methods inevitably neglect valuable correlations from pixels outside the fixed neighborhood. We solve the problem ingeniously through embedding the input feature into a query feature map with full resolution and multiple reference feature maps with downscaled resolutions. In such a manner, the computation burden can be greatly relieved when applying our method in low-level pixel-wise image processing tasks.

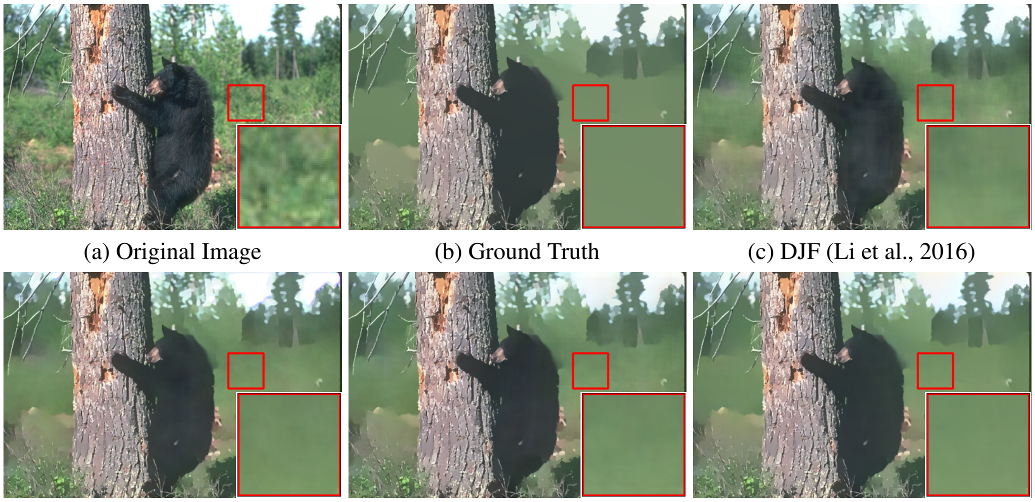
Third, the pyramid non-local block can be easily incorporated into existing CNN-based models devised for low-level image processing, such as RDN(Zhang et al., 2018) for denoising or MemNet (Tai et al., 2017b) for single image super-resolution. Experiments on various datasets indicate that the pyramid non-local block consistently improves the performance in image smoothing, denoising and super-resolution.

4 Experiments in Edge-preserving Smoothing

Edge-preserving smoothing is a fundamental topic in image processing. It aims at detecting major image structures while neglecting insignificant details, which is critical in many computer vision tasks such as image segmentation and contour detection.

Table 1: Quantitative comparison in terms of PSNR/SSIM on three smoothing filters. The best performance is marked in bold. Model depth and parameter number are also indicated.

	DJF (Li et al., 2016)	CEILNet (Fan et al., 2017)	ResNet (Zhu et al., 2019)	PNEN
WMF	34.31/0.9647	37.73/0.9773	38.30/0.9813	**39.45/0.9846**
L0	30.20/0.9458	31.30/0.9519	32.30/0.9671	**33.44/0.9741**
SD Filter	30.95/0.9264	32.67/0.9452	33.21/0.9532	**34.19/0.9646**
Max depth	6	32	37	37
#Parameters	99k	1113k	1961k	1875k



(a) Original Image (b) Ground Truth (c) DJF (Li et al., 2016)

(d) CEILNet (Fan et al., 2017) (e) ResNet (Zhu et al., 2019) (f) Our PNEN

Figure 4: Visual comparison of learning $L0$ smoothing filter. There exist unwanted details in previous methods if we examine closely. In contrast, the result generated by our PNEN is closer to the groundtruth.

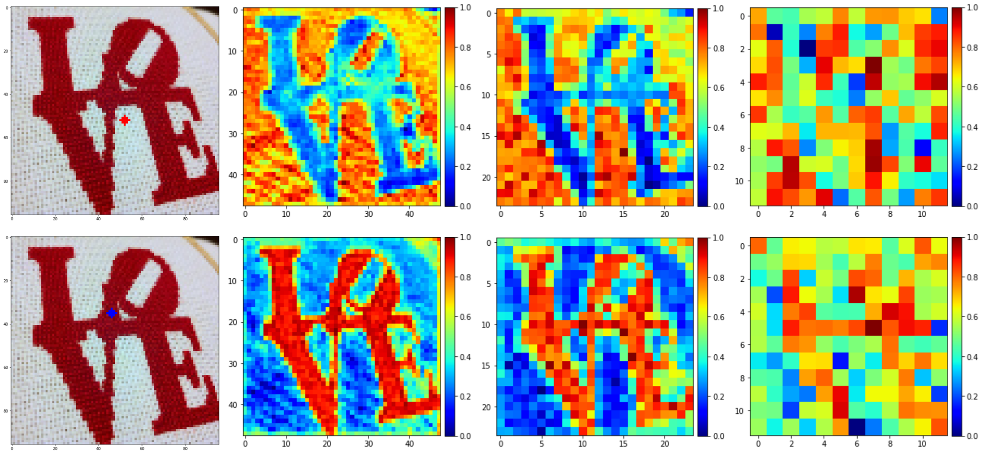
4.1 Dataset

The image datasets are from Zhu et al. (2019). There are 500 images of clear structures and visible details to evaluate the learning apability of deep model in reproducing edge-preserving filters. Following Zhu et al. (2019), we use 400 images as training set and the remaining 100 images as testing set. We train models to reproduce three representative filters in our experiments, including weighted median filter ($r = 10, \sigma = 50$) (Zhang et al., 2014), L_0 smoothing ($\lambda = 0.02, \kappa = 2$) (Xu et al., 2011) and SD Filter ($\lambda = 15$) (Ham et al., 2017).

Implementation Details Without specification, all convolutional layers have 64 filters with kernel size 3×3. We stack three PNB-s and DRB-s consecutively as the feature extractor, resulting in a 37-layer deep network. During the training stage, random horizontal flip and rotation are applied for data augmentation. Training images are split into 96×96 patches. The mini-batch size is set to 8. Adam optimizer (Kingma & Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ is used for optimization. The initial learning rate is set to 5×10^{-4} and reduced by half when training loss stops decreasing, until it reduces to 10^{-4}. It takes around 2 days to train a model on one TITAN Xp GPU. For inference, the model takes 1.2 seconds to process a testing image of 500×400 pixels.

4.2 Comparison with the State-of-the-art

Recently, piles of CNN-based approaches (Xu et al., 2015; Liu et al., 2016; Li et al., 2016; Fan et al., 2017; Zhu et al., 2019) have been proposed to reproduce edge-preserving smoothing filters. We compare our proposed PNEN against three state-of-the-art methods, including DeepJointFilter (DJF) (Li et al., 2016), Cascaded Edge and Image Learning Network (CEILNet) (Fan et al., 2017) and Residual Networks (ResNet) (Zhu et al., 2019). For fair comparison, the networks are re-trained from



(a) Input Image (b) Scale 2 (c) Scale 4 (d) Scale 8

Figure 5: Correlation map of pyramid non-local operation at different scales. The first row shows the correlation map computed at the red point. The second row shows correlation map computed at the blue point.

Table 2: Ablation study on different design choices of pyramid non-local block.

	1	2	3	4	5	6
Scale 2		✓			✓	✓
Scale 4			✓		✓	✓
Scale 8				✓		✓
PSNR	31.95	32.57	33.17	32.87	33.21	**33.44**
SSIM	0.9665	0.9706	0.9708	0.9707	0.9720	**0.9741**

Table 3: Comparison of computation resource requirements in terms of floating point operations (FLOPs) and memory consumption.

	w/o	w/ Wang et al. (2018)	w/ our PNB
FLOPs	40.6G	46.2G	42.5G
Memory	1.1GB	9.6GB	4.3GB
#Parameters	1334k	1519k	1875k
PSNR	31.95	32.40	33.44

scratch on the same training dataset as described above. We evaluate the quality of the generated images using two metrics, including Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index (SSIM) (Wang et al., 2004).

Quantitative results are reported in Table 1. When reproducing WMF, L_0 smoothing and SD Filter, our proposed method (PNEB) outperforms the second best method ResNet (Zhu et al., 2019) by 1.15dB, 1.14dB and 0.98dB in PSNR metric respectively. Higher SSIM performance also indicates that our method can give rise to smoothed images with better structural information. A visual comparison of learning L_0 smoothing filter is provided in Figure 4. As can be seen in close-ups of the selected patch, the region smoothed by our proposed PNEN appears to be cleaner and flatter than other methods. Our result is closer to the groundtruth image.

As shown in Figure 5, we visualize the similarity map derived from the last pyramid non-local block at two locations, marked by red and blue respectively. Figure 5 (b), (c) and (d) show the similarity map at 'Scale 2', 'Scale 4' and 'Scale 8', respectively. We can see that the pixels with similar features show high correlation in these maps. Thus, the features are non-locally enhanced by exploiting long-range dependencies. Particularly, for the pixel marked by red, there exist noises when estimating correlation coefficients with similar patterns (left-bottom area) in 'Scale 2' while 'Scale 4' performs well on these patterns. It indicates that the adoption of pyramid non-local operations can benefit the robustness of estimating correlations between different scales of patterns.

4.3 Ablation Study

To validate the effectiveness and necessity of pyramidal strategy, we exhaustively compare PNB with its variants on learning $L0$ smoothing filter. The performance changes in terms of PSNR and SSIM are shown in Table 2. The multi-scale pyramid non-local networks outperforms single-scale non-local

Table 4: Image denoising results. Average PSNR/SSIM results are reported on Set12, BSD68 and Urban100 dataset.

Dataset	RDN	RDN(deeper)	RDN(w/ PNB)
Set12	30.60/0.8651	30.62/0.8662	**30.72/0.8689**
BSD68	29.30/0.8335	29.31/0.8341	**29.38/0.8370**
Urban100	30.28/0.8923	30.32/0.8945	**30.83/0.9023**
#Parmeters	20.5M	23.2M	20.9M

Table 5: SISR results. Average PSNR/SSIM results are reported on Set5, Set14, BSD100 and Urban100 dataset.

Dataset	MemNet	MemNet(deeper)	MemNet(w/ PNB)
Set5	34.09/0.9248	34.12/0.9250	**34.18/0.9267**
Set14	30.00/0.8350	30.04/0.8361	**30.12/0.8409**
BSD100	28.96/0.8001	28.97/0.8008	**29.02/0.8072**
Urban100	27.56/0.8376	27.65/0.8380	**27.88/0.8474**
#Parmeters	677k	1056k	1047k

networks with a significant margin. It is noteworthy that the design of PNB is flexible according to the trade-off between computation efficiency and accuracy.

The required computational resources, in terms of floating point operations (FLOPs) and memory consumption, are summarized in Table 3. The performance is obtained by testing methods on a 96×96 patch. We compare our proposed PNB with the non-local block proposed in Wang et al. (2018). Memory consumption increases dramatically since non-local operation needs to store a large correlation matrix. Our memory requirement is 57% less than that of Wang et al. (2018). Under the same GPU memory constraint, our method allows larger training patches and larger receptive field for low-level image processing tasks.

5 Extension: Experiments on Image Restoration

Deep convolutional networks are widely used in image restoration tasks (Kim et al., 2016a;b; Zhang et al., 2017; Lim et al., 2017; Tai et al., 2017b; Hui et al., 2018; Zhang et al., 2018). We adopt two state-of-the-art methods, RDN (Zhang et al., 2018) and MemNet (Tai et al., 2017b), as the baseline models for image denoising and image super-resolution task, respectively. As discussed in Section 3.4, efficient computation allows our proposed pyramid non-local block be incorporated into these low-level baseline models. The PNB acts as a basic component to exploit non-local image self-similarity. Model architectures and visual comparisons are provided in the appendix.

Image Denoising: Three PNB-s are incorporated into RDN. To demonstrate the effectiveness of PNB, we compare with a variant of RDN by stacking more convolutional layers. The deeper RDN has more parameters than the PNB-enhanced RDN. We follow the training and testing protocols as in Zhang et al. (2018) for fair comparison. 800 images from DIV2K dataset (Timofte et al., 2017) are utilized as training data. We add white Gaussian noise with standard deviation $\sigma = 25$ to the original images to synthesize noisy images. Results on three widely used benchmarks, Set12 (Zhang et al., 2017), BSD68 (Roth & Black, 2009), Urban 100 (Huang et al., 2015), are reported in Table 4. The PNB-enhanced RDN yields better results than original RDN and its variant.

Image Super-Resolution: We add three PNB-s into MemNet. We also compare with a variant of MemNet by stacking more convolutional layers. The deeper MemNet and PNB-enhanced MemNet have similar parameter numbers. We follow the training and testing protocols as in Tai et al. (2017b) for fair comparison. The training sets consist of 291 images where 200 images are BSD train set and other 91 images are from Yang et al. (2010). The training data are constructed by bicubic downsampling with factor 3, and then upscaled to the original size. Results on four widely used benchmarks, Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), BSD100 (Martin et al., 2001) and Urban100 (Huang et al., 2015), are presented in Table 5. The PNB-enhanced MemNet yields better results than original MemNet and its variant.

6 Conclusion

In this paper, we present a novel and effective pyramid non-local enhanced network (PNEN) for edge-preserving image smoothing. The proposed pyramid non-local block (PNB) is computation-friendly, which allows it be a plug-and-play component in existing deep methods for low-level image processing tasks. Methods incorporating our proposed pyramid non-local block achieve significantly improved performance in image denoising and image super-resolution.

REFERENCES

Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 252–268, 2018.

Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 60–65. IEEE, 2005.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug 2007. ISSN 1057-7149.

Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

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.

Bumsub Ham, Minsu Cho, and Jean Ponce. Robust guided image filtering using nonconvex potentials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5197–5206, 2015.

Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 723–731, 2018.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1646–1654, 2016a.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1637–1645, 2016b.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3587–3596, 2017.

Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. Non-locally enhanced encoder-decoder network for single image de-raining. *arXiv preprint arXiv:1808.01491*, 2018a.

Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 517–532, 2018b.

Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *European Conference on Computer Vision*, pp. 154–169. Springer, 2016.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 136–144, 2017.

Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pp. 1673–1682, 2018.

Sifei Liu, Jinshan Pan, and Ming-Hsuan Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *European Conference on Computer Vision*, pp. 560–576. Springer, 2016.

Julien Mairal, Francis R Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, volume 29, pp. 54–62. Citeseer, 2009.

Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pp. 2802–2810, 2016.

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.

Stefan Roth and Michael J Black. Fields of experts. *International Journal of Computer Vision*, 82(2): 205, 2009.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.

Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of International Conference on Computer Vision*, 2017b.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 114–125, 2017.

Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4799–4807, 2017.

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.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l 0 gradient minimization. In *ACM Transactions on Graphics (TOG)*, volume 30, pp. 174. ACM, 2011.

Li Xu, Jimmy Ren, Qiong Yan, Renjie Liao, and Jiaya Jia. Deep edge-aware filters. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1669–1678, 2015.

Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472–480, 2017.

Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pp. 711–730. Springer, 2010.

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*, 2017.

Qi Zhang, Li Xu, and Jiaya Jia. 100+ times faster weighted median filter (wmf). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2830–2837, 2014.

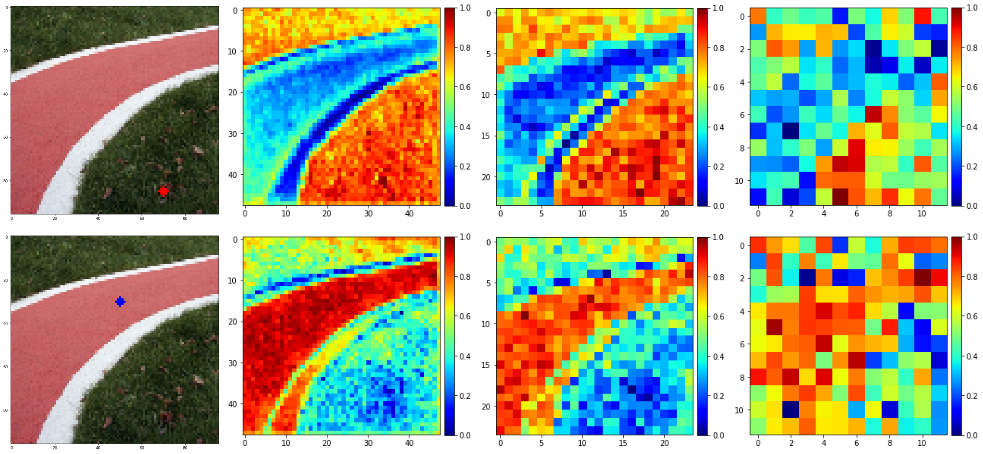
Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *arXiv preprint arXiv:1812.10477*, 2018.

Feida Zhu, Zhetong Liang, Xixi Jia, Lei Zhang, and Yizhou Yu. A benchmark for edge-preserving image smoothing. *IEEE Transactions on Image Processing*, 2019.

A Correlation Map Visualization

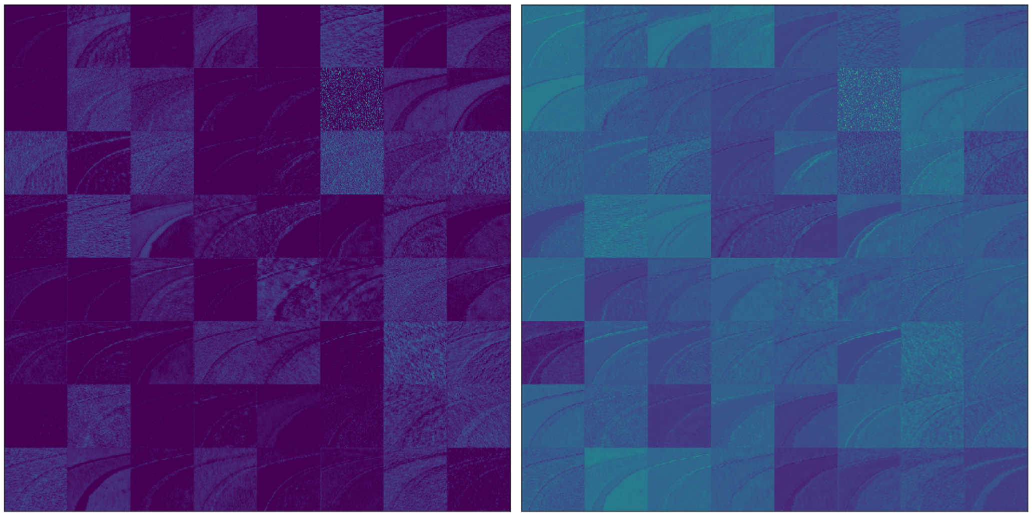
Here, we visualize correlation maps computed by the pyramid non-local block (PNB). In each input image, the correlation maps at two positions, marked by red and blue point, are shown respectively. As shown in Figure 6 and Figure 8, we can see that the pixels with similar features show high correlation in the correlation map.

The 64 channel feature activation maps, before and after applying PNB, are visualized in Figure 7 and Figure 9. They are placed in the 8 × 8 grid. The features are non-locally enhanced by exploiting image self-similarity, showing stronger structure information after applying pyramid non-local block.



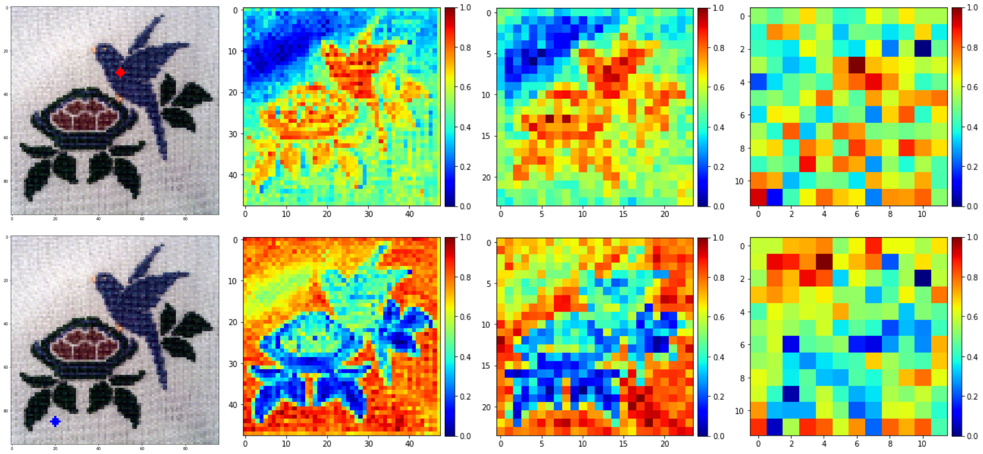
Input Image Scale 2 Scale 4 Scale 8

Figure 6: Correlation map of pyramid non-local operation at different scales. The first row shows the correlation map computed at the red point. The second row shows correlation map computed at the blue point.



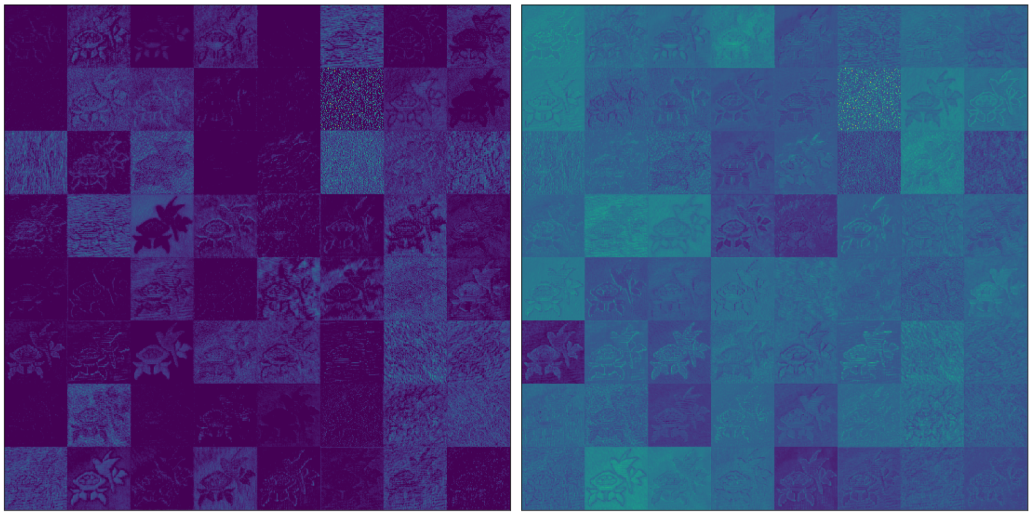
Feature Map before PNB Feature Map after PNB

Figure 7: Feature activation maps before and after applying pyramid non-local block.



Input Image Scale 2 Scale 4 Scale 8

Figure 8: Correlation map of pyramid non-local operation at different scales. The first row shows the correlation map computed at the red point. The second row shows correlation map computed at the blue point.

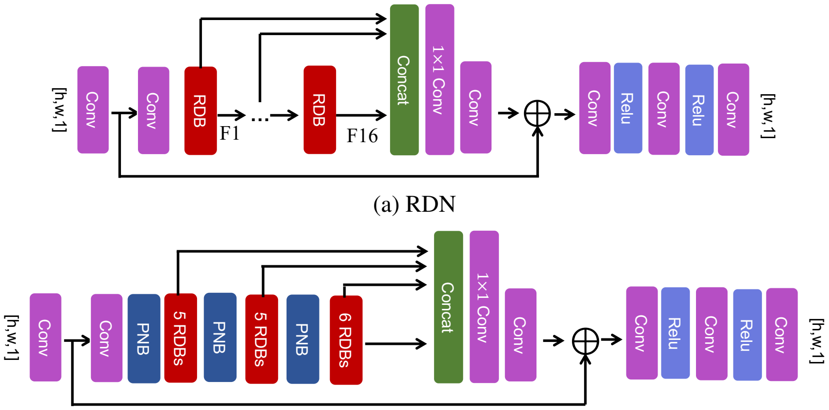


Feature Map before PNB Feature Map after PNB

Figure 9: Feature activation maps before and after applying pyramid non-local block.

B Image Denoising: Models and Visual Comparison

The architecture of RDN (Zhang et al., 2018) is shown in Figure 10 (a), which is built upon EDSR (Lim et al., 2017). They replaced residual blocks with their proposed residual dense block (RDB). Please refer to Zhang et al. (2018) for more details. We incorporate three our proposed PNB-s into their network every five RDBs, as shown in Figure 10 (b).

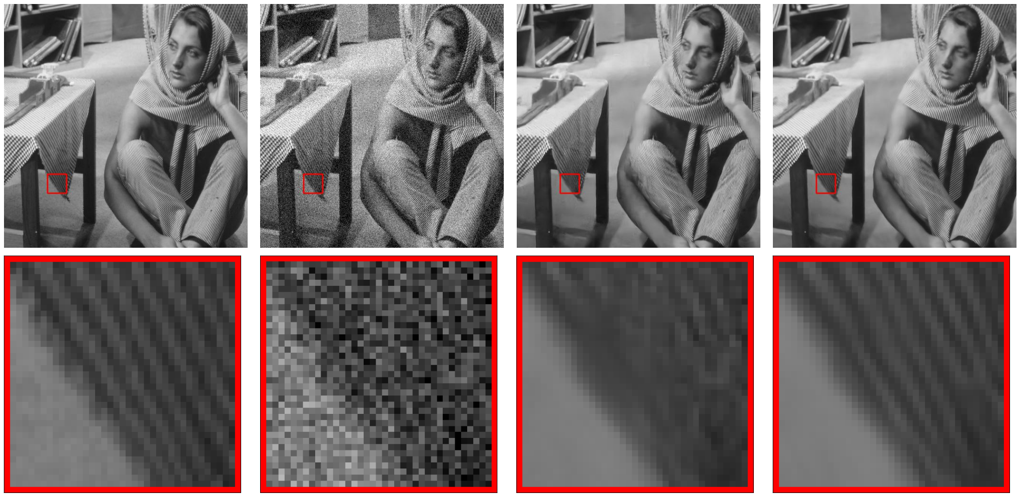


(a) RDN

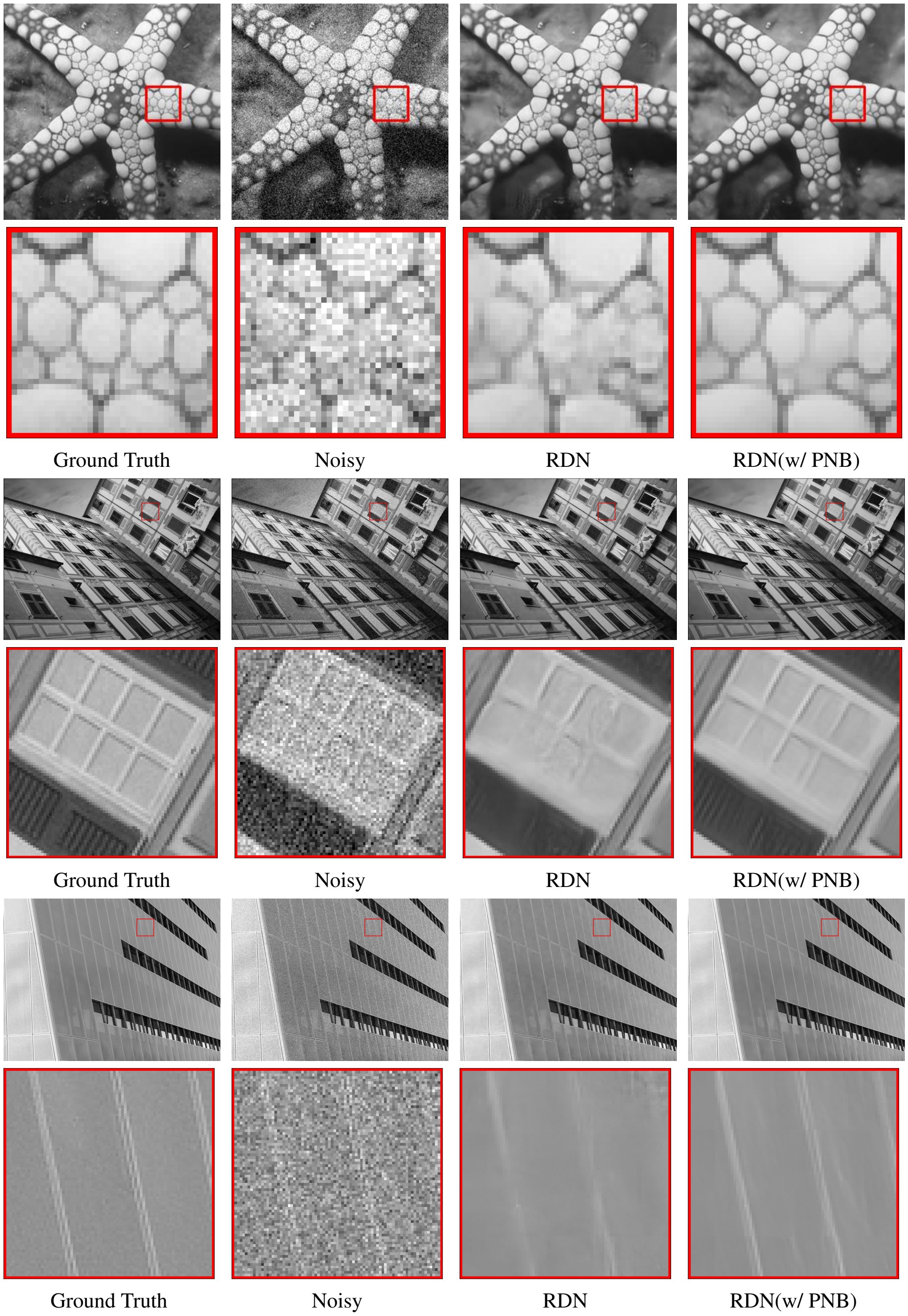
(b) RDN w/ PNB

Figure 10: Baseline RDN architecture, and the enhanced RDN architecture by incorporating our PNB.

Here, we show more qualitative results of RDN and the PNB-enhanced RDN. The groundtruth clean images are degraded with white Gaussian noise (level: $\sigma = 25$) to synthesize noisy images. From left to right, they are the groundtruth clean images, noisy images, results generated by RDN and results generated by RDN with PNB, respectively.



Ground Truth Noisy RDN RDN(w/ PNB)



Ground Truth Noisy RDN RDN(w/ PNB)

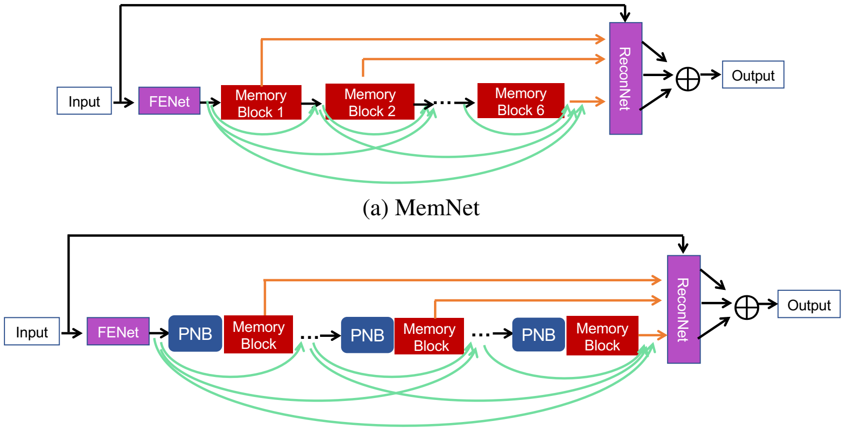
Ground Truth Noisy RDN RDN(w/ PNB)

Ground Truth Noisy RDN RDN(w/ PNB)

Figure 11: Denoising Comparison. From left to right, they are the ground truth clean images, noisy images, results generated by RDN and results generated by RDN with PNB, respectively.

C Image Super-resolution: Models and Visual Comparison

The architecture of MemNet (Tai et al., 2017b) is shown in Figure 12 (a). They proposed memory block as their basic component. Please refer to Tai et al. (2017b) for more details. As shown in Figure 12 (b), we incorporate three our proposed PNB-s into their network without affecting the dense connections between memory blocks.

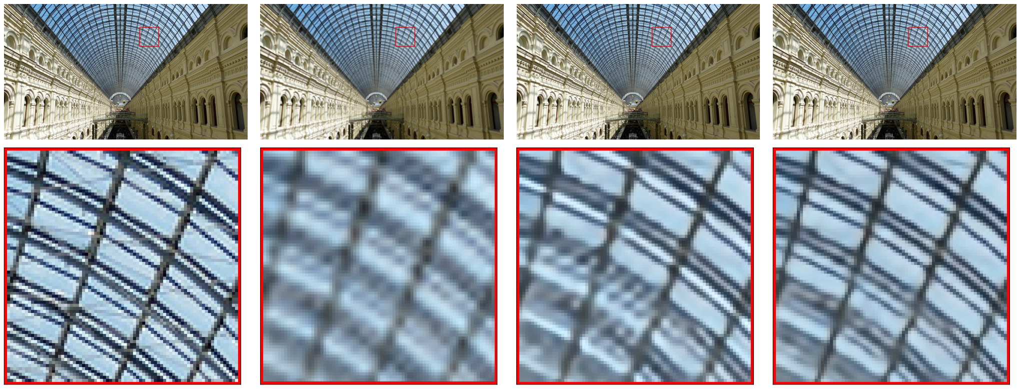


(a) MemNet

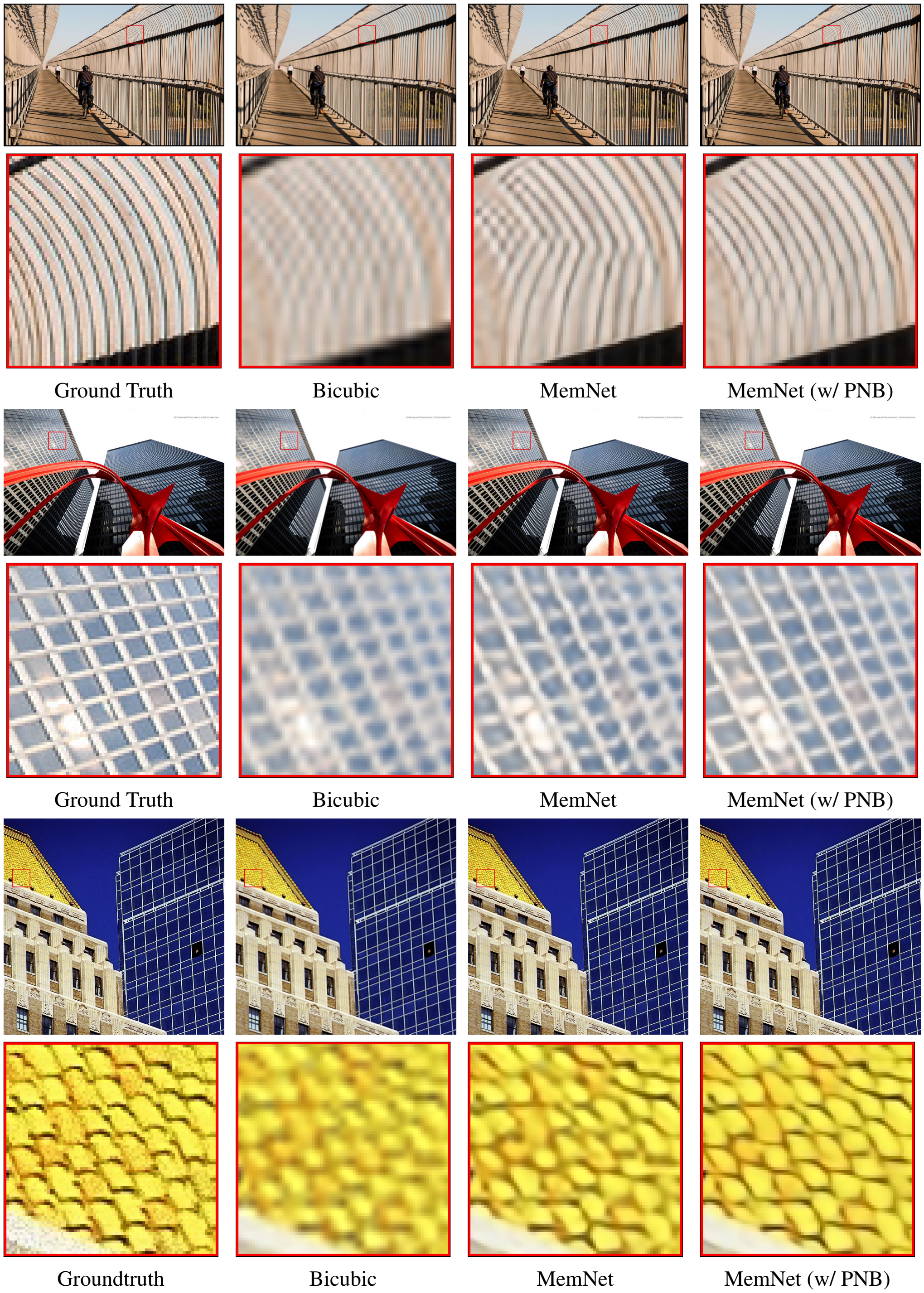
(b) MemNet w/ PNB

Figure 12: Baseline MemNet architecture, and the enhanced MemNet architecture by incorporating our PNB.

We show more qualitative results of MemNet and the PNB-enhanced MemNet. The groundtruth high-resolution (HR) images are bicubic downsampled and upsampled by factor 3 to synthesize low-resolution (LR) images. From left to right, they are the groundtruth HR images, bicubic-interpolated images, results generated by MemNet and results generated by MemNet with PNB, respectively.



Ground Truth Bicubic MemNet MemNet (w/ PNB)



Ground Truth Bicubic MemNet MemNet (w/ PNB)

Ground Truth Bicubic MemNet MemNet (w/ PNB)

Groundtruth Bicubic MemNet MemNet (w/ PNB)

Figure 13: Super-resolution Comparison. From left to right, they are the groundtruth high-resolution images, bicubic-interpolated images, results generated by MemNet and results generated by MemNet with PNB, respectively.