Adaptive Consistency Prior based Deep Network for Image Denoising

Chao Ren* Xiaohai He Chuncheng Wang Zhibo Zhao
College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China
{chaoren, hxh}@scu.edu.cn, wangchuncheng@stu.scu.edu.cn, mzhibozhao@gmail.com

Abstract

Recent studies have shown that deep networks can achieve promising results for image denoising. However, how to simultaneously incorporate the valuable achievements of traditional methods into the network design and improve network interpretability is still an open problem. To solve this problem, we propose a novel model-based denoising method to inform the design of our denoising network. First, by introducing a non-linear filtering operator, a reliability matrix, and a high-dimensional feature transformation function into the traditional consistency prior, we propose a novel adaptive consistency prior (ACP). Second, by incorporating the ACP term into the maximum a posteriori framework, a model-based denoising method is proposed. This method is further used to inform the network design, leading to a novel end-to-end trainable and interpretable deep denoising network, called DeamNet. Note that the unfolding process leads to a promising module called dual element-wise attention mechanism (DEAM) module. To the best of our knowledge, both our ACP constraint and DEAM module have not been reported in the previous literature. Extensive experiments verify the superiority of DeamNet on both synthetic and real noisy image datasets.

1. Introduction

In computer vision applications, images corrupted by noise will significantly affect the further analysis and process. Generally, representative image denoising methods can be categorized as filtering-based methods [48, 23, 4, 11, 15, 29, 26], model-based methods [30, 42, 14, 39, 56, 19, 55, 54, 57, 37, 38, 58, 7], and learning-based methods [14, 33, 61, 34, 31, 32, 41, 22, 18, 6, 17, 3, 24].

Filtering-Based Methods. Classical filtering-based methods, *e.g*., median filtering [48] and Wiener filtering [23], *etc*., exploit certain manually designed low pass filters to remove the image noise. Knaus *et al*. [26] proposed to progressively reduce noise by deterministic annealing with a simple iterative filtering scheme. In addition, based on the observation that a natural image patch generally has many similar counterparts within the same image, some works remove the noise by using a stack of non-local similar patches, such as the classic non-local means (NLM) algorithm [4], and the block-matching based 3D filtering (BM3D) algorithm [11]. Because of the success of NLM and BM3D, various non-local filtering methods appeared [15, 29]. However, the block-wise operation may lead to blurred outputs. Moreover, the difficulty of setting the hyper-parameters of these methods will significantly affect the performance.

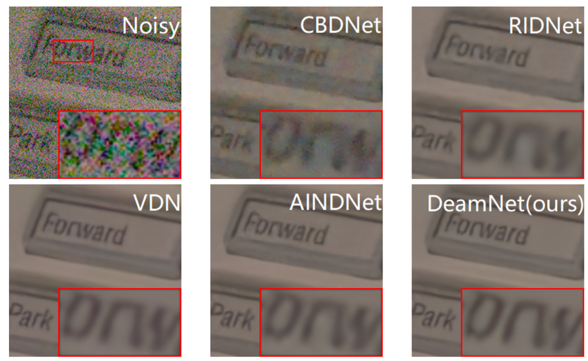

Figure 1. A real noisy image from SIDD dataset [1]. Compared with other denoising methods, DeamNet achieves better results.

Model-Based Methods. The model-based methods usually formulate the denoising task as a maximum a posteriori (MAP)-based optimization problem, whose performance mainly depends on image priors. With the assumption that an image patch can be sparsely represented by some proper basis function, the sparsity prior [42, 14] was proposed for image restoration (IR). Recently, a trilateral weighted sparse coding scheme based image denoising method was proposed for real-world images in [56]. With the observation that a matrix formed by the nonlocal similar patches in a natural image has a low-rank property, the low-rank prior based image denoising methods [19, 55, 57, 54] were proposed. For instance, Xu *et al*. [19] proposed a weighted nuclear norm minimization (WNNM) method for IR via a low-rank matrix approximation. In addition, Pang *et al*. [37] introduced graph-based regularizers to reduce image

*Corresponding author.

noise. Although these model-based methods have strong mathematical derivations, the performance will be significantly degraded in recovering texture structures for heavy noise. Furthermore, they are usually time-consuming since the high complexity of the iterative optimization.

Learning-Based Methods. The learning-based methods focus on learning a latent mapping from the noisy image to the clean version, and can be divided into traditional learning-based [14, 9, 46, 33, 36] and deep network-based methods [31, 62, 41, 22, 18, 6, 17, 5, 13]. Recently, because the deep network-based methods have achieved more promising denoising results than the filtering-based, model-based, and traditional learning-based methods, they have become the dominant approaches. For example, Zhang *et al*. [61] proposed a simple but effective denoising convolutional neural network (CNN) by stacking convolution, batch normalization and rectified linear unit (ReLU) layers. Inspired by the non-local similarity of images, the non-local operations were incorporated into a recurrent neural network in [31] for IR. A neural nearest neighbors block (N3 block) based architecture was proposed in [41]. Chang *et al*. [5] proposed a novel spatial-adaptive denoising network for efficient single image noise removal. For real-noise denoising, Anwar *et al*. [3] proposed a one-stage denoising network with feature attention, and Kim *et al*. [24] proposed a well-generalized denoising architecture and a transfer learning scheme. However, the architectures of most deep network-based methods are empirically designed and the achievements of the traditional algorithms are not fully considered, facing weak interpretability to some extent.

More recently, some deep unfolding-based methods [9, 13, 59, 22, 28] were proposed to implement some traditional methods by deep networks. For example, Chen *et al*. [9] implemented the classic iterative nonlinear reaction diffusion method as a deep network for IR. Based on the fractional optimal control theory, Jia *et al*. [22] developed a fractional optimal control network for image denoising. Lefkimmiatis *et al*. [28] utilized the proximal gradient method (PGM) to solve a constrained optimization problem transformed by the general MAP optimization problem in IR, and the PGM updates were implemented as a neural network.

Contributions. The main focus of this work is to propose a novel model-based denoising method, and then exploit the inference process of this method to inform the design of our denoising CNN (DeamNet). The details about the differences among DeamNet and the existing denoising networks are presented in the 'Supplementary Material'.

Our main contributions are as follows: 1) A novel image prior (*i.e*., adaptive consistency prior (ACP)) is proposed to improve the traditional consistency prior by introducing a non-linear filtering operator, a reliability matrix, and a high-dimensional transformation function. Then, a model-based denoising method is proposed by exploiting ACP under the MAP framework; 2) the iterative optimization steps of the model-based denoising method are utilized to inform the network design, leading to an end-to-end trainable and interpretable denoising network (DeamNet). DeamNet merges the power of model-based scheme with deep learning; 3) the unfolding process leads to a novel and promising module called dual element-wise attention mechanism (DEAM) module, which is derived from the reliability matrix in ACP. DEAM also enables the across-level/across-scale feature interactions and the element-wise feature recalibration in each iteration stage; 4) a multi-scale nonlinear operation (NLO) sub-network with DEAM modules is proposed, which can simultaneously exploit the fine scale and coarse scale features within the NLO sub-network for better nonlinear filtering in feature domain (FD). To the best of our knowledge, we are the first to propose the ACP constraint and DEAM module. Experiments verify the effectiveness of DeamNet for both synthetic and real image denoising.

2. Proposed Method

We propose a novel ACP-based denoising framework, which is further used to inform the design of our DeamNet.

2.1. Adaptive Consistency Prior for Denoising

Because of the local continuity and non-local self-similarity of natural images, strong correlations are prone to hold locally and non-locally. Based on these correlations, the consistency priors were proposed [60, 43, 8, 35].

Limitations of Consistency Prior and Their Solutions. Let $\mathbf{x} \in \mathbb{R}^n$ be an image with n pixels, x_i represent the i-th pixel in $\mathbf{x}$, $\mathcal{D}_i$ denote the index vector for the related pixels of x_i within $\mathbf{x}$, and w_{ij} be the normalized weight related to x_i and x_j. The consistency prior $\mathcal{J}_{\text{CP}}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^1$ can be written as

$$\mathcal{J}_{\text{CP}}(\mathbf{x}) \overset{\text{def}}{=} \sum_{i=1}^{n} \|x_i - \sum_{j\in\mathcal{D}_i} w_{ij}x_j\|_2^2. \tag{1}$$

To facilitate the analysis, we rewrite Eq. (1) as follows:

$$\mathcal{J}_{\text{CP}}(\mathbf{x}) = \| \underset{③}{\underline{\underline{\mathbf{I}}}} (\underset{①}{\underline{\underline{\mathbf{x}}}} - \underset{②}{\underline{\underline{\mathbf{W}}}}\, \underset{①}{\underline{\underline{\mathbf{x}}}})\|_2^2, \tag{2}$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is an identical matrix, and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a diagonal consistency matrix composed by w_{ij}-s.

We can understand the consistency prior as follows: the image $\mathbf{x}$ in pixel domain (labeled by ①) is first filtered by the linear consistency matrix $\mathbf{W}$ (labeled by ②), and then the magnitude of the fitting deviation vector $(\mathbf{x} - \mathbf{W}\mathbf{x})$ is uniformly constrained by $\mathbf{I}$ (labeled by ③). Therefore, the main limitations of the consistency prior and their solutions are: 1) the consistency prior performs a constraint in pixel domain. However, according to the traditional methods, *e.g*., BM3D and its variants [10, 11], performing denoising in FD can better reconstruct image details. In addition, by

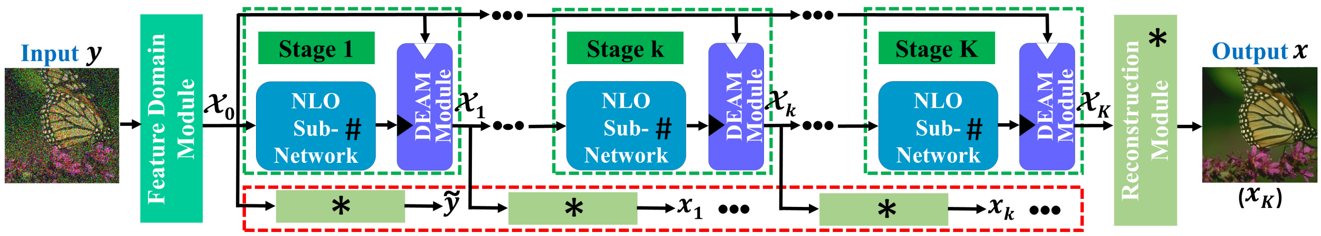

Figure 2. Architecture of the proposed DeamNet. It consists of a feature domain (FD) module, a reconstruction module, and K iteration stages based on nonlinear operation (NLO) sub-networks and dual element-wise attention mechanism (DEAM) modules. The modules labeled by $*$ mean the parameters of these modules are shared. The parameters of the modules labeled by $\#$ are also shared.

expanding the space dimension, more features can be captured for better details recovery. Thus, it is promising to use a high-dimensional transformation function $\mathscr{T}(\cdot)$ on $\mathbf{x}$; 2) the linear operation for each pixel by $\mathbf{W}$ may oversmooth image details, leading to poor performance. For better edge and image details preservation properties, it is desired to introduce an adaptive and non-linear filtering operator $\mathscr{K}(\cdot)$ to replace $\mathbf{W}$; 3) the fitting deviations are penalized uniformly in the consistency prior. However, it would be useful to adaptively penalize each fitting deviation according to the reliability of the corresponding pixels. This motivates using a reliability matrix $\mathbf{\Lambda}$ to adaptively weight $(\mathbf{x} - \mathbf{W}\mathbf{x})$.

Proposed Adaptive Consistency Prior (ACP). The previous analyses motivate the proposing of ACP, which integrates the concepts of FD, non-linear filtering, and reliability estimation. Let $\mathscr{T}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\cdot m}$ denote one transformation function, $\mathbf{\Lambda} = \mathscr{D}(a_1, ..., a_l, ..., a_{nm}) \in \mathbb{R}^{nm\times nm}$ be the diagonal reliability matrix with elements a_l-s on the main diagonal ($a_l > 0$), and $\mathscr{K}(\cdot) : \mathbb{R}^{n\cdot m} \to \mathbb{R}^{n\cdot m}$ represent one nonlinear filtering operator. ACP can be written as

$$\mathcal{J}_{\text{ACP}}(\mathbf{x}|\underbrace{\mathscr{T}, \mathscr{K}, \mathbf{\Lambda}}_{\text{pre-specified}}) \overset{\text{def}}{=} \|\mathbf{\Lambda}(\mathscr{T}(\mathbf{x}) - \mathscr{K}(\mathscr{T}(\mathbf{x})))\|_2^2. \quad (3)$$

There are some interesting special cases for different settings of $\mathcal{J}_{\text{ACP}}(\cdot)$. For example, $\mathcal{J}_{\text{ACP}}(\mathbf{x}|\mathbf{I}, \mathscr{K}, \mathbf{\Lambda}) = \|\mathbf{\Lambda}(\mathbf{x} - \mathscr{K}(\mathbf{x}))\|_2^2$ becomes ACP in the pixel domain; $\mathcal{J}_{\text{ACP}}(\mathbf{x}|\mathbf{I}, \mathbf{W}, \mathbf{I}) = \mathcal{J}_{\text{CP}}(\mathbf{x})$ becomes the original consistency prior. In other words, the consistency prior is a special case of ACP, and the function space of $\mathcal{J}_{\text{CP}}(\cdot)$ is expanded for modelling complex constraint relationships in $\mathcal{J}_{\text{ACP}}(\cdot)$.

Proposed Denoising Optimization Problem. Let $\mathcal{X} = \mathscr{T}(\mathbf{x})$ and $\mathcal{X}_k + \Delta\mathcal{X} = \mathcal{X}$, then $\mathscr{K}(\mathcal{X}_k + \Delta\mathcal{X})$ can be approximated using Taylor series around the k-th iteration:

$$\mathscr{K}(\mathcal{X}_k + \Delta\mathcal{X}) \approx \mathscr{K}(\mathcal{X}_k) + \mathbf{J}_k \Delta\mathcal{X}, \quad (4)$$

where $\mathbf{J}_k$ is the Jacobian matrix, and thus we can obtain

$$\begin{aligned}&\mathcal{J}_{\text{ACP}}(\mathbf{x}|\mathscr{T}, \mathscr{K}, \mathbf{\Lambda}) \approx \|\mathbf{\Lambda}(\mathcal{X} - \mathscr{K}(\mathcal{X}_k))\|_2^2 \\ &+ \|\mathbf{\Lambda}(\mathbf{J}_k\Delta\mathcal{X})\|_2^2 - 2(\mathcal{X} - \mathscr{K}(\mathcal{X}_k))^T \mathbf{\Lambda}^T \mathbf{\Lambda} \mathbf{J}_k \Delta\mathcal{X},\end{aligned} \quad (5)$$

where the second and third terms tend to zero for a small perturbation $\Delta\mathcal{X}$. When $\mathcal{X}$ is in the vicinity of $\mathcal{X}_k$, we can get an approximated ACP $\mathcal{J}^{\star}_{\text{ACP}}(\mathbf{x}|\mathscr{T}, \mathscr{K}, \mathbf{\Lambda}) = \|\mathbf{\Lambda}(\mathcal{X} - \mathscr{K}(\mathcal{X}_k))\|_2^2$. By incorporating the approximated ACP and the data fidelity term $\Psi(\mathbf{x}|\mathbf{y}, \mathscr{T}) = \|\mathcal{Y} - \mathcal{X}\|_2^2$ ($\mathcal{Y} = \mathscr{T}(\mathbf{y})$), we can obtain the novel ACP-driven denoising algorithm:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \Psi(\mathbf{x}|\mathbf{y}, \mathscr{T}) + \lambda \mathcal{J}^{\star}_{\text{ACP}}(\mathbf{x}|\mathscr{T}, \mathscr{K}, \mathbf{\Lambda}), \quad (6)$$

where λ is the regularization parameter ($\lambda > 0$). Since $\mathscr{K}(\cdot)$, $\mathbf{\Lambda}$, λ, and $\mathscr{T}(\cdot)$ are preset before the optimization, $\mathbf{x}$ is the only unknown variable that needs to be estimated.

Note that in the original ACP, $\mathscr{K}(\mathscr{T}(\mathbf{x}))$ is relevant to the optimization variable $\mathbf{x}$ (*i.e.*, $\mathscr{K}(\mathscr{T}(\mathbf{x}))$ is signal-dependent) and it is a highly nonlinear operator. Although the original ACP driven denoising problem can be potentially solved by using the gradient descent method, the gradient of $\mathscr{K}(\cdot)$ (denoted by $\triangledown\mathscr{K}(\cdot)$) is difficult to be computed because of its highly nonlinear property. By using the approximated ACP, $\mathbf{x}$ is replaced by the k-th estimate $\mathbf{x}_k$ during the iteration, which avoids the calculation of $\triangledown\mathscr{K}(\cdot)$. The convergence of the approximated ACP-based method can be guaranteed by forcing $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 < \|\mathbf{x}_k - \mathbf{x}\|_2^2$ ($\mathbf{x}$ is the ground-truth image). This will be achieved by designing a loss constraint illustrated in the next subsection.

Theorem 1. *Let $\beta_l = 1/(1 + \lambda a_l^2) \in (0, 1)$, $\boldsymbol{\beta} \in \mathbb{R}^{n\cdot m}$ be the tensor form of all $\{\beta_l\}$-s, $\mathscr{L}(\cdot) : \mathbb{R}^{n\cdot m} \to \mathbb{R}^n$ be the reconstructing operator from FD to the pixel domain, $\mathbb{1}$ denote a tensor of all ones with the same size as $\boldsymbol{\beta}$, and $\otimes$ be the element-wise product of two tensors. Then, the solution of the ACP-driven denoising problem in Eq. (6) can be obtained by*

$$\mathbf{x}_{k+1} = \mathscr{L}(\boldsymbol{\beta} \otimes \mathscr{T}(\mathbf{y}) + (\mathbb{1} - \boldsymbol{\beta}) \otimes \mathscr{K}(\mathscr{T}(\mathbf{x}_k))). \quad (7)$$

Proof. Since Eq. (6) is a quadratic optimization problem, by constraining the derivative of Eq. (6) to 0, we can easily obtain the iterative equation in Eq. (7). More details about the proof are provided in the 'Supplementary Material'. □

To accurately estimate the clean image $\mathbf{x}$, how to predesign the best $\{\mathscr{K}(\cdot), \mathbf{\Lambda}, \lambda, \mathscr{T}(\cdot), \mathscr{L}(\cdot)\}$ is the key issue. Since the deep learning-based methods are continuously showing superiority over traditional model-based methods, this paper proposes an end-to-end trainable unfolding network which adaptively learns these operators.

2.2. Deep Unfolding Denoising Network

Manually designing $\{\mathscr{K}(\cdot), \mathbf{\Lambda}, \lambda, \mathscr{T}(\cdot), \mathscr{L}(\cdot)\}$ for high performance is very challenging and time-consuming.

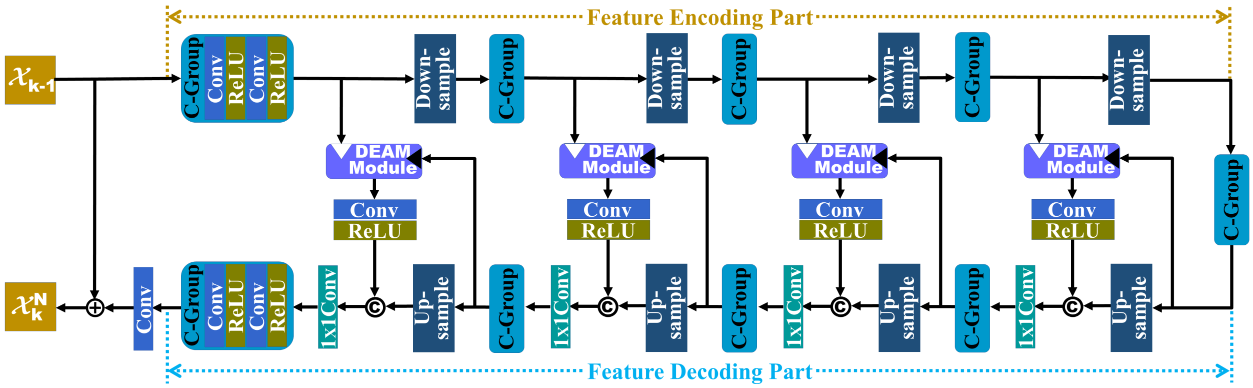

Figure 3. Architecture of the NLO sub-network, where ⓒ denotes Concatenation. It mainly contains three parts: feature encoding part (FEP, a series of Conv groups (C-Groups) followed by downsampling layers), feature decoding part (FDP, a series of C-Groups followed by upsampling layers), and DEAM module group (for across-scale features recalibration and interaction).

Therefore, the proposed model-based framework is implemented via one deep unfolding denoising network constructed by the FD module ($\mathscr{T}(\cdot)$), reconstruction module ($\mathscr{L}(\cdot)$), NLO sub-network ($\mathscr{K}(\cdot)$), and DEAM module (λ and $\mathbf{\Lambda}$). Let $\mathbf{\Theta}_\psi$ be the trainable parameter set for an operator ψ. In our network, the hyper-parameters λ, $\mathbf{\Lambda}$, $\mathbf{\Theta}_\mathscr{T}$, $\mathbf{\Theta}_\mathscr{K}$, and $\mathbf{\Theta}_\mathscr{L}$ are learned in a discriminative manner.

Network Architecture. DeamNet shown in Fig. 2 is a trainable and extended version of the ACP-driven denoising problem. It contains K iterative stages in FD. First, the learned version of the FD operator $\mathscr{T}(\cdot)$ is applied to the noisy image $\mathbf{y}$ to obtain an initial feature $\mathcal{Y} = \mathscr{T}(\mathbf{y})$. Second, $\mathcal{Y}$ is fed into a series of encoder-decoder architecture based NLO sub-networks, whose parameters are shared. These sub-networks are the learned version of $\mathscr{K}(\cdot)$. The input and output of the k-th NLO sub-network are denoted as $\mathcal{X}_{k-1}$ and $\mathcal{X}_k^{\mathrm{N}}$ (note that $\mathcal{X}_0 = \mathcal{Y}$), and $\mathcal{X}_k^{\mathrm{N}} = \mathscr{K}(\mathcal{X}_{k-1})$. Next, since $\boldsymbol{\beta} \otimes \mathcal{Y} + (\mathbb{1} - \boldsymbol{\beta}) \otimes \mathscr{K}(\mathcal{X}_k)$ in Eq. (7) is closely related to the attention mechanism, to calculate the dual weights ($\boldsymbol{\beta}$, $\mathbb{1} - \boldsymbol{\beta}$) and perform the dual summation, DEAM module is proposed. Specifically, $\mathcal{Y}$ and $\mathcal{X}_k^{\mathrm{N}}$ are both input into the DEAM module to obtain the recalibrated features in the k-th stage, which ensures the availability of low-level information in the long CNN. The output of the k-th DEAM module can be written as $\mathcal{X}_k = \mathcal{F}_{\mathrm{deam}}^k([\mathcal{X}_k^{\mathrm{N}}, \mathcal{Y}])$, where $\mathcal{F}_{\mathrm{deam}}^k(\cdot)$ denotes the k-th DEAM operator.

Finally, the output $\mathcal{X}_k$ will be reconstructed by the reconstruction module to obtain the k-th image estimate $\mathbf{x}_k = \mathscr{L}(\mathcal{X}_k)$. Note that the parameters of all the reconstruction modules are shared. To guarantee the convergence of Eq. (7), the proposed network should have a good self-correcting ability (*i.e.*, $\|\mathbf{x}_K - \mathbf{x}\|_2^2 < \|\mathbf{x}_{K-1} - \mathbf{x}\|_2^2 < \ldots < \|\mathbf{x}_1 - \mathbf{x}\|_2^2$ should hold), motivating a multi-stage loss function. Moreover, to guarantee $\mathscr{L}(\cdot)$ is the inversion version of $\mathscr{T}(\cdot)$, we add an extra branch composed by the reconstruction module right after the FD module, and then force the output of the branch to be closed to the input $\mathbf{y}$. With N clean-noisy training pairs $\{\mathbf{x}(g), \mathbf{y}(g)\}_{g=1}^N$, our network can be trained by optimizing the following L_p loss function:

$$\begin{aligned}\mathcal{L}(\mathbf{\Theta}) &= \frac{1}{KN}\sum_{k=1}^{K}\sum_{g=1}^{N}\xi_k\|\mathcal{F}_{\mathrm{DeamNet}}^k(\mathbf{y}(g)) - \mathbf{x}(g)\|_p^p \\ &+ \frac{1}{N}\sum_{g=1}^{N}\eta\|\mathscr{L}(\mathscr{T}(\mathbf{y}(g))) - \mathbf{y}(g)\|_p^p,\end{aligned} \tag{8}$$

where $\sum_{k=1}^K \xi_k = 1$ and $0 < \xi_1 < \xi_2 < \ldots < \xi_K < 1$. $\mathbf{\Theta} = [\boldsymbol{\beta}, \mathbf{\Theta}_\mathscr{T}, \mathbf{\Theta}_\mathscr{K}, \mathbf{\Theta}_\mathscr{L}]$ denotes the trainable parameter set ($\boldsymbol{\beta}$ is related to λ and $\mathbf{\Lambda}$). $\mathcal{F}_{\mathrm{DeamNet}}^k(\cdot)$ denotes the mapping function of the k-th stage of DeamNet from the noisy image to the clean version. By simply setting $\xi_k = \vartheta\xi_{k-1}$ ($\vartheta > 1$), we can get $\xi_k = \frac{\vartheta - 1}{\vartheta^K - 1}\vartheta^{k-1}$. In general, the L_2 loss has good confidence in Gaussian noise, whereas the L_1 loss has better tolerance for outliers. Therefore, we set $p = 2$ for Gaussian noise and $p = 1$ for real noise.

Denoising in High-Dimensional FD. The FD operator is used to imitate the process of $\mathscr{T}(\cdot)$, which projects the noisy image $\mathbf{y}$ to a high-dimensional FD to get an initial feature estimate $\mathcal{Y} = \mathscr{T}(\mathbf{y})$. It is implemented by a simple structure: a Conv layer followed by a residual unit where an ReLU layer is put between two Conv layers. After K iteration stages, the output $\mathcal{X}_K$ is reconstructed by the reconstruction module $\mathscr{L}(\cdot)$, which is implemented by a simple structure: a residual unit where an ReLU layer is put between two Conv layers followed by a Conv layer with 1 filter. Finally, we can obtain the estimate $\mathscr{L}(\mathcal{X}_K)$. To guarantee $\mathscr{L}(\cdot)$ be the inverse operator of $\mathscr{T}(\cdot)$, a branch only composed by the FD and reconstruction modules is added, and the output of the branch is forced to be the same as the input.

Performing denoising in high-dimensional FD has the following advantages: 1) the original noisy space can be potentially transformed to an FD space where noise can be more easily reduced, leading to finer image details than the pixel domain. Experiments in the 'Supplementary Material' show this in details; 2) using high-dimensional FD module can also increase the feature channel number and improve

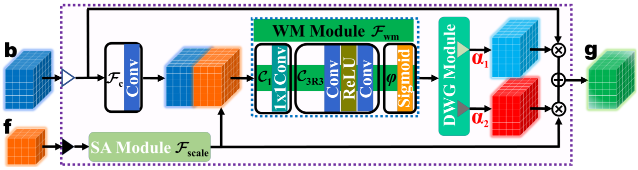

Figure 4. The diagram of the DEAM module. $\otimes$ denotes the element-wise product. The scale adjusted (SA) module upscales $\mathbf{f}$ to the same spatial resolution of $\mathbf{b}$. The weights mapping (WM) module generates the weighting tensor. The dual weights generator (DWG) module generates two dual weighting tensors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ ($\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2 = \mathbb{1}$) for $\mathbf{b}$ and $\mathbf{f}$, respectively.

the information flow transmission in the deep network, leading to higher performance. In contrast, the traditional deep unfolding networks perform iterative denoising in the low-dimensional image space; 3) by regarding the residual between the ground-truth image and the degraded image as noise to be reduced, this strategy can also make the network more useful for other IR applications.

NLO Sub-Network. Inspired by the success of U-Net [44], we model $\mathscr{K}(\cdot)$ by an encoder-decoder architecture with T spatial resolution scales as shown in Fig. 3. Specifically, the input feature $\mathcal{X}_{k-1}$ is progressively filtered and down-sampled (down-sampling is performed by one Conv layer with stride of 2) in the feature encoding part (FEP) for greatly expanding the receptive field. Accordingly, the down-sampled features are progressively filtered and up-sampled (up-sampling is implemented by a combination of $\times 2$ interpolation and one Conv layer with stride of 1) in the feature decoding part (FDP). To allow the adaptive feature recalibration and across-scale feature interaction for better network expressive ability, the DEAM module is introduced into the sub-network. The recalibrated and interacted features are then adjusted by a Conv layer followed by ReLU, and then fused with the up-sampled Conv group (C-Group) features via Concat and a 1×1 Conv layer. Overall, the t-th DEAM-related process in each NLO sub-network can be regarded as an across-scale fusion operator on the output $\mathcal{X}_k^{et}$ of the t-th C-Group in FEP and the output $\mathcal{X}_k^{\mathrm{d}(t+1)}$ of the C-Group at the $(t+1)$-th scale in FDP. Finally, the residual reconstruction operator $\mathcal{F}_{\text{r-rec}}(\cdot)$ (implemented by a Conv layer) is used to reconstruct the residual features, and thus the output of the NLO sub-network can be written as

$$\mathcal{X}_k^{\mathrm{N}} = \mathscr{K}(\mathcal{X}_{k-1}) = \mathcal{F}_{\text{r-rec}}(\mathcal{X}_k^{\mathrm{d1}}) + \mathcal{X}_{k-1}. \tag{9}$$

DEAM Module. How to obtain the dual weights ($\boldsymbol{\beta}$ and $1-\boldsymbol{\beta}$) and implement the element-wise product of $\mathcal{Y}$ and $\mathscr{K}(\mathcal{X}_k)$ in Eq. (7) are crucial for the unfolding denoising network. By regarding $\mathcal{Y}$ as the low-resolution (LR) branch, $\mathscr{K}(\mathcal{X}_k)$ as the high-resolution (HR) branch, and $\{\boldsymbol{\beta}, 1-\boldsymbol{\beta}\}$ as the attention maps for the LR and HR branches, we can see that Eq. (7) is closely related to the attention mechanism. To make our module has better adaptivity, we introduce a scale adjusted (SA) module for across-level/across-scale features interactions. All of these motivate the proposing of our DEAM module as shown in Fig. 4.

Specifically, DEAM module has two inputs (coarse-level feature $\mathbf{b}$ and high-level feature $\mathbf{f}$) and one output $\mathbf{g}$. First, $\mathbf{b}$ is adjusted by using a Conv layer $\mathcal{F}_{\mathrm{c}}(\cdot)$, and $\mathbf{f}$ is processed by using the SA module $\mathcal{F}_{\text{scale}}(\cdot)$. In our network, the SA module is an identical matrix for the output of the NLO sub-network in each stage, and is the Up-sample module for the output of the C-Group in FDP within the NLO sub-network. Then, these two adjusted inputs are concatenated by a Concat layer to obtain the feature $\mathbf{f}_0 = [\mathcal{F}_{\mathrm{c}}(\mathbf{b}), \mathcal{F}_{\text{scale}}(\mathbf{f})]$. After that, $\mathbf{f}_0$ is sent into the weights mapping (WM) module. In the WM module, a 1×1 Conv layer $\mathcal{C}_1$ is first used to reduce the dimension of $\mathbf{f}_0$. Next, two 3×3 Conv layers with s_0 and s channels ($s_0 < s$) and a ReLU layer are used to generate the initial element-wise feature weights for both stability and nonlinearity. A sigmoid activation layer φ is used to normalize the weights to $(0,\ 1)$ and generate the weighting tensor $\boldsymbol{\alpha}$. Overall, $\boldsymbol{\alpha}$ can be written as

$$\boldsymbol{\alpha} = \mathcal{F}_{\text{wm}}(\mathbf{f}_0) = \varphi(\mathcal{C}_{3\mathrm{R}3}(\mathcal{C}_1(\mathbf{f}_0))), \tag{10}$$

where $\mathcal{F}_{\text{wm}}(\cdot)$ represents the WM operator, and $\mathcal{C}_{3\mathrm{R}3}$ denotes the operator of two 3×3 Conv layers with one ReLU. Then, $\boldsymbol{\alpha}$ is input into the dual weights generator (DWG) module to generate two dual weighting tensors (*i.e.*, $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}_2 = \mathbb{1} - \boldsymbol{\alpha}$) for $\mathbf{b}$ and $\mathbf{f}$, respectively. Finally, the output of the DEAM module can be formulated as

$$\mathbf{g} = \boldsymbol{\alpha}_1 \otimes \mathbf{b} + \boldsymbol{\alpha}_2 \otimes \mathbf{f} = \boldsymbol{\alpha} \otimes \mathbf{b} + (\mathbb{1} - \boldsymbol{\alpha}) \otimes \mathbf{f}. \tag{11}$$

In the DEAM module of each iteration stage, $\boldsymbol{\alpha} = \boldsymbol{\beta}$ and $\mathbb{1} - \boldsymbol{\alpha} = \mathbb{1} - \boldsymbol{\beta}$ are used for weighting $\mathcal{Y}$ and $\mathscr{K}(\mathcal{X}_k)$. In the DEAM module of each NLO sub-network, $\boldsymbol{\alpha}$ and $\mathbb{1} - \boldsymbol{\alpha}$ are used for weighting $\mathcal{X}_k^{et}$ and $\mathcal{X}_k^{\mathrm{d}(t+1)}$.

The advantages of DEAM are discussed in the following. As we know, the attention mechanism [63] enables the network to have discriminative ability for different types of information. However, most attention mechanisms in IR do not pay much attention to the interactions of across-level/across-scale features. For example, in the residual architecture, the lower-frequency information at the coarse-level (low-frequency branch) represents the principal component of the input. However, the importance of low-frequency branch is largely ignored in traditional attention mechanisms. In contrast, DEAM can adaptively increase the feature weight for residual branch if the low-frequency branch is not ideal in describing the latent features, and vice versa. In addition, the initial feature information availability and across-level feature interactions in DEAM can further enhance the network's expressive ability.

2.3. Further Analyze and Optimize DeamNet

DeamNet is a trainable and extended version of the ACP-driven denoising problem, which explains its effectiveness in a mathematical way to some extent. However, many sub-branches with reconstruction modules and loss functions are used in Eq. (8). It not only makes the network training difficult but also restricts the freedom of the network parameters. In addition, it is challenging and time-consuming to preset the parameters ξ_k-s and η in Eq. (8). To make the network architecture and training more compact, the added reconstruction modules labeled by the red rectangle in Fig. 2 are removed, and thus the following optimized loss function scheme is adopted instead of the original scheme:

$$\mathcal{L}(\mathbf{\Theta}) = \frac{1}{N}\sum_{g=1}^{N}\|\mathcal{F}^{K}_{\text{DeamNet}}(\mathbf{y}(g)) - \mathbf{x}(g)\|_p^p. \quad (12)$$

We compare these two schemes on addition white Gaussian noise (AWGN) with noise level 15, 25, 50 on Urban100 [21]. For the original scheme, ϑ and η are empirically set to 3 and 0.2 for best performance. Other network settings are provided in subsection 3.1. Results (Peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM)) in Table 1 show that the optimized version achieves slightly better results than the original one. This may be caused by the difficult training of the original scheme since too many constraints are used, while the optimized scheme has much more freedom for pursuing better fitting ability. Therefore, the optimized scheme is used as our default scheme.

3. Experimental Results

In this section, we demonstrate the effectiveness of our method on both synthetic and real noisy datasets. **Due to the limited space, more experimental results and further analysis are given in the 'Supplementary Material'.** Code will be available at https://github.com/chaoren88/DeamNet.

3.1. Network Implementation Details

The Pytorch framework is used to train DeamNet with a GeForce GTX 1080Ti GPU. We empirically set the iteration stage number K to 4 (for the speed-accuracy trade-off), the scale T in NLO to 5, and the size of all Conv layers to 3×3 except for those layers right after concatenation layers with kernel size 1×1. Moreover, all the Conv layers have 64 filters, except for that in the channel-downscaling (4 filters) and the reconstruction layer (1 filters for a grayscale image and 3 filters for a color image). During the training, network parameters are first initialized by the Xavier approach [16], and then optimized by the Adam [25] with default settings. In each training batch, eight patches are extracted as the input. We initially set the learning rate to 10^{-4}, and then fine-tune the network with the learning rate 10^{-5}.

3.2. Dataset Preparation and Testing

To train DeamNet for the synthetic AWGN, the Berkeley Segmentation Dataset (BSD) [12] and Div2K [51] are adopted. The clean images are corrupted by AWGN with

Table 1. PSNR (dB) and SSIM results of original scheme and optimized scheme on Urban100.

Noise level	15	25	50
Original Scheme	33.30/0.9365	30.84/0.9042	27.45/0.8362
Optimized Scheme	33.37/0.9372	30.85/0.9048	27.53/0.8373

Table 2. PSNRs (dB) and SSIMs on Urban100 for varying K.

K	1	2	3	4	5
PSNR/SSIM	30.57/0.8992	30.73/0.9023	30.81/0.9041	30.85/0.9048	30.89/0.9054

Table 3. Ablation investigation for DeamNet. Average PSNR (dB) and SSIM values on Urban100 for noise level 25.

FD	×	×	✓	✓
DEAM	×	✓	×	✓
PSNR/SSIM	30.01/0.8890	30.52/0.8990	30.65/0.9018	30.85/0.9048

specific levels 15, 25 and 50. We evaluate the denoising performance on three standard benchmarks including Set12 [61], BSD68 [45] and Urban100 [21]. PSNR and SSIM [64] metrics are used for objective evaluation. For real noisy images, we use the SIDD [1] and RENOIR [2] datasets for training. In both synthetic and real noise cases, we randomly crop these training image pairs into small patches of size 128×128. To augment training samples, rotation of 180° and horizontal flipping are applied. Furthermore, we adopt three datasets for denoising on real-world images:

• **DnD** [40] is composed of 50 real-world noisy images, but the near noise-free counterparts are not available. Fortunately, the PSNR/SSIM results can be achieved by uploading the denoised images to the DnD website.

• **SIDD** [1] provides 320 pairs of noisy images and the near noise-free counterparts for training and 1280 image patches for validation. The PSNR/SSIM results can be obtained by submitting the denoised images to the SIDD website.

• **RNI15** [27] provides 15 real-world noisy images. Unfortunately, the ground-truth clean images are unavailable, therefore we only present the visual results for RNI15.

3.3. Study of Parameter K

The selection of the iteration stage number K is crucial for DeamNet, and thus the effects of different settings of K are tested. Specifically, five network models with $K = 1, 2, 3, 4, 5$ are trained independently for noise level $\sigma = 25$, and then compared with each other in the experiment. Table 2 reports the average PSNR and SSIM values on Urban100. From the results, we can conclude that even only using one stage, our network can achieve promising denoising performance. Furthermore, with the increasing of K, the performance becomes even better. For example, the PSNR/SSIM gains of $K = 4$ over $K = 1, 2, 3$ are 0.28dB/0.0056, 0.12dB/0.0025, and 0.04dB/0.0007, respectively. When K increases from 4 to 5, 0.04dB/0006 PSNR/SSIM gains can still be obtained. That means $K = 4$ is a good choice for $\sigma = 25$. Note that, slightly better results can be obtained by setting a larger K for a larger σ. In our imple-

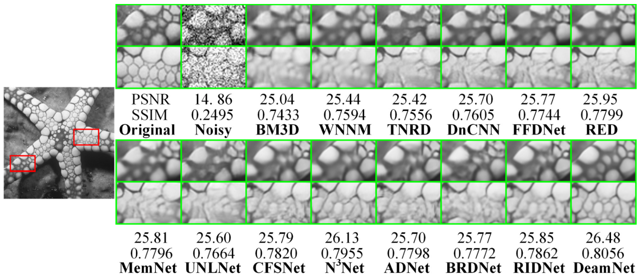

Figure 5. Visual quality comparison for 'Starfish' from Set12.

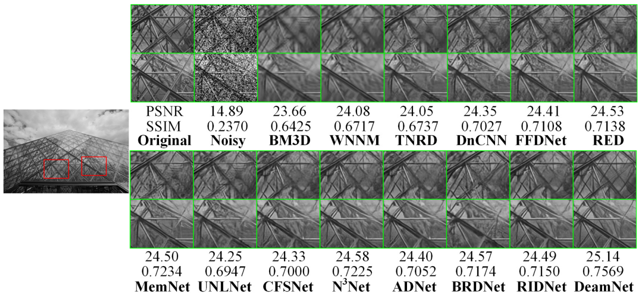

Figure 6. Visual quality comparison for 'test044' from BSD68.

mentation, K is set to 4 to best balance the performance and the network complexity for various noise levels.

3.4. Ablation Study

Study of FD. To show the effect of FD, we remove the FD module from DeamNet. That means denoising is performed in pixel domain, and the input and output of each iteration stage and NLO sub-network have only one channel. The results in Table 3 show that the performance will significantly decrease without using FD. For instance, compared the third column with the first column, the performance decreases from 30.65dB/0.9018 to 30.01dB/0.8890 without using FD. These results verify that the FD processing is more effective than the pixel domain processing in DeamNet.

Study of DEAM. From Table 3, we can conclude that DEAM is an effective module in DeamNet, leading to higher performance. For example, by adding the DEAM module, the PSNR/SSIM gains are 0.51dB/0.0100 for the network without FD. These comparisons show that the DEAM modules are essential for the performance of DeamNet.

Note that the network without FD and DEAM modules can be regarded as an extended version of the traditional consistency prior based denoising method. The results in Table 3 show that the ACP based denoising network (using FD and DEAM modules) obtains much better results than the traditional consistency prior based network, which verify the superiority of ACP over the consistency prior.

3.5. Denoising on Synthetic Noisy Images

To evaluate the performance of DeamNet, 13 state-of-the-art and representative denoising methods are tested. These baselines include one filtering-based method BM3D [11], one model-based method WNNM [19], and 11 deep network-based methods TNRD [9], DnCNN [61], FFDNet [62], RED [34], MemNet [47], UNLNet [28], CFSNet [53], N^3Net [41], ADNet [49], BRDNet [50], and RIDNet [3]. Moreover, the self-ensemble [52] results denoted by the super script * is also presented to maximize potential denoising performance of DeamNet. The average PSNR/SSIM results of the competing methods are shown in Table 4, and the perceptual comparisons are shown in Figs. 5 and 6.

Table 4. Quantitative comparison results of the competing methods with noise level 15, 25 and 50 on Set12, BSD68 and Urban100.

Method	15			25			50		
	Set12	BSD68	Urban100	Set12	BSD68	Urban100	Set12	BSD68	Urban100
BM3D[11]	32.37 0.8952	31.07 0.8717	32.35 0.9220	29.97 0.8504	28.57 0.8013	29.70 0.8777	26.72 0.7676	25.62 0.6864	25.95 0.7791
WNNM[19]	32.70 0.8982	31.37 0.8766	32.97 0.9271	30.28 0.8577	28.83 0.8087	30.39 0.8885	27.05 0.7775	25.87 0.6982	26.83 0.8047
TNRD[9]	32.50 0.8958	31.42 0.8769	31.86 0.9031	30.06 0.8512	28.92 0.8093	29.25 0.8473	26.82 0.768	25.97 0.6994	25.88 0.7563
DnCNN[61]	32.86 0.9031	31.73 0.8907	32.68 0.9255	30.44 0.8622	29.23 0.8278	29.97 0.8797	27.18 0.7829	26.23 0.7189	26.28 0.7874
FFDNet[62]	32.75 0.9027	31.63 0.8902	32.43 0.9273	30.43 0.8634	29.19 0.8289	29.92 0.8886	27.32 0.7903	26.29 0.7245	26.52 0.8057
RED[34]	- -	- -	- -	- -	- -	- -	27.34 0.7897	26.35 0.7245	26.48 0.7991
MemNet[47]	- -	- -	- -	- -	- -	- -	27.38 0.7933	26.35 0.7297	26.64 0.8029
UNLNet[28]	32.69 0.9001	31.47 0.8858	32.47 0.9252	30.27 0.8576	28.96 0.8197	29.80 0.8831	27.07 0.7793	26.04 0.7129	26.14 0.7911
CFSNet[53]	32.48 0.8905	31.29 0.8694	32.12 0.9138	30.44 0.8623	29.24 0.8290	29.91 0.8848	27.22 0.7855	26.28 0.7206	26.36 0.7934
N^3Net[41]	- -	- -	- -	30.50 0.8651	29.30 0.8329	30.19 0.8910	27.43 0.7950	26.39 0.7302	26.82 0.8141
ADNet[49]	32.98 0.9050	31.74 0.8916	32.87 0.9308	30.58 0.8654	29.25 0.8294	30.24 0.8923	27.37 0.7908	26.29 0.7216	26.64 0.8073
BRDNet[50]	33.03 0.9055	31.79 0.8926	33.02 0.9322	30.61 0.8657	29.29 0.8309	30.37 0.8934	27.45 0.7935	26.36 0.7265	26.82 0.8131
RIDNet[3]	32.91 0.9059	31.81 0.8934	33.11 0.9339	30.60 0.8672	29.34 0.8331	30.49 0.8975	27.43 0.7932	26.40 0.7267	26.73 0.8132
DeamNet	**33.19** **0.9097**	**31.91** **0.8957**	**33.37** **0.9372**	**30.81** **0.8717**	**29.44** **0.8373**	**30.85** **0.9048**	**27.74** **0.8057**	**26.54** **0.7368**	**27.53** **0.8373**
DeamNet*	**33.21** **0.9098**	**31.93** **0.8959**	**33.45** **0.9375**	**30.86** **0.8720**	**29.46** **0.8374**	**30.95** **0.9056**	**27.81** **0.8067**	**26.57** **0.7370**	**27.66** **0.8400**

We can observe that DeamNet achieves the highest average PSNR and SSIM values. Take the case of noise level 50 on Set12, BSD68, and Urban100 as examples. For one of the most representative model-based method WNNM[19], the improvement is about 0.7dB, and the SSIM gain is about 0.028∼0.038. Compared with the classic deep CNN based method DnCNN[61], our method can achieve PSNR gain about 0.3∼1.3dB, and SSIM gain about 0.018∼0.050. DeamNet also outperforms the nonlocal self-similarity based denoising networks UNLNet[28] and N^3Net[41] by large margins on all the test datasets. Especially, the PSNR/SSIM gains are about 1.39dB/0.0462 and 0.71dB/0.0232 on Urban100, respectively. Even compared with the ADNet [49], BRDNet [50], and RIDNet[3], our method outperforms them at all noise levels on these datasets. Furthermore, the perceptual results of two images with noise level 50 show that DeamNet is able to reconstruct the clear structures of the noisy images. In contrast, the comparison baselines may overly smooth the fine image details. For example, the details are finer in the red rectangle region of each image recovered by DeamNet. For better visual comparisons, these red rectangle regions are zoomed-in. According to the results, the superiority of DeamNet are

Table 5. Real image denoising results of several state-of-the-art algorithms on DnD and SIDD benchmark datasets.

Dataset	CBM3D	TNRD	DnCNN	FFDNet	CBDNet	VDN	RIDNet	AINDNet(TF)	DeamNet	DeamNet*
DnD	34.51 0.8507	33.65 0.8306	32.43 0.7900	37.61 0.9415	38.06 0.9421	39.38 0.9518	39.25 0.9528	39.37 0.9505	**39.63** **0.9531**	**39.70** **0.9535**
SIDD	25.65 0.685	24.73 0.643	23.66 0.583	29.30 0.694	33.28 0.868	39.26 0.955	37.87 0.943	38.95 0.952	**39.35** **0.955**	**39.43** **0.956**

Table 6. Running time (in seconds) and parameter comparisons.

Method	BM3D	WNNM	TNRD	DnCNN	FFDNet	RED	MemNet	CFSNet	N^3Net
Size 256^2	0.76	210.26	0.47	0.01	0.01	1.36	0.88	0.04	0.17
Size 512^2	3.12	858.04	1.33	0.05	0.03	4.70	3.61	0.11	0.74
Size 1024^2	12.82	3603.68	4.61	0.16	0.11	15.77	14.69	0.38	3.25
# Params	-	-	27k	558k	490k	4131k	667k	1731k	706k
Method	ADNet	BRDNet	CBM3D	CBDNet	VDN	RIDNet	AINDNet	Ours	-
Size 256^2	0.02	0.05	0.98	0.03	0.04	0.07	0.05	0.05	-
Size 512^2	0.06	0.20	4.63	0.06	0.07	0.21	0.21	0.19	-
Size 1024^2	0.20	0.76	22.85	0.25	0.19	0.84	0.80	0.73	-
# Params	519k	1115k	-	6793k	7817k	1499k	13764k	2225k	-

demonstrated both objectively and subjectively.

3.6. Denoising on Real Noisy Images

Because the real noises are usually signal dependent and spatially variant according to different in-camera pipelines, real image denoising is generally a highly challenging task. To further show the generalization ability of DeamNet for real noises, DnD benchmark [40], SIDD benchmark [1], and RNI15 dataset [27] are chosen as test datasets. Note that, for DnD and SIDD benchmarks, the near noise-free images are not publicly available, but the PSNR/SSIM results can be obtained through their online servers. While for RNI15, only the noisy images are available. Several state-of-the-art denoising methods that have demonstrated their validity on real noisy images are tested for comparisons, including CBM3D [10], TNRD [9], DnCNN [61], FFDNet [62], CBDNet [20], VDN [59], RIDNet [3], and AINDNet(TF) [24]. According to [24], AINDNet uses different datasets to train several models. AINDNet(S) is trained with the synthetic dataset. Although it performs well on DnD, its performance on SIDD is very low. AINDNet(TF) updates specified parameters from AINDNet(S) with real noisy images and the best overall performance can be gotten on both DnD and SIDD, and thus AINDNet(TF) is used for a fair comparison. The results of different methods are provided in Table 5 and Figs. 7, 8, and 9. We can conclude that DeamNet consistently achieves the best performance on these datasets among all the competing methods, including the newly proposed denoising methods for real noisy images, *i.e.*, CBDNet, VDN, RIDNet, and AINDNet(TF). Furthermore, the visual results show that our method removes noises robustly, suppresses artifacts effectively, and preserves image edges well. Overall, the superiority of DeamNet on real noisy image denoising confirms the effectiveness of our network design.

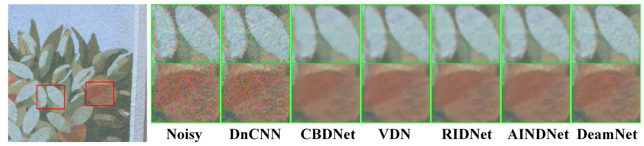

Figure 7. Visual comparison of real denoising results from DnD.

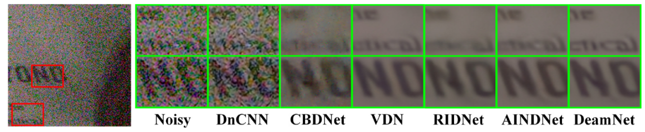

Figure 8. Visual comparison of real denoising results from SIDD.

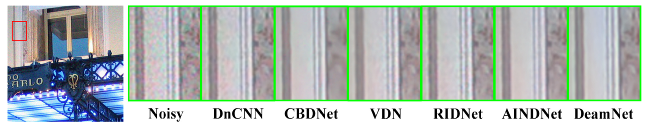

Figure 9. Visual comparison of real denoising results from RNI15.

3.7. Computational Complexity

To make a comparison in computational complexity, both the network parameter numbers and the average running times on the GPU (except for BM3D[11], CBM3D [10], WNNM[19], and TNRD [9], which are on the CPU) of different methods for images of size 256×256, 512×512, and 1024×1024 are shown in Table 6. Although DeamNet is slower than DnCNN [61], FFDNet [62], CFSNet [53], ADNet [49], and CBDNet [10], its performance is significantly better. DeamNet also outperforms BM3D[11], CBM3D [10], WNNM[19], RED [34], MemNet [47], and N^3Net [41] by a considerable margin with a lower running time. Furthermore, when compared with BRDNet [50], RIDNet [3] and AINDNet[24], our DeamNet can still obtain higher PSNRs with a slightly lower running time. For the parameters, DeamNet has a reasonable parameter number, which is significantly lower than RED [34], CBDNet [10], VDN [59], and AINDNet [24]. Consequently, the effectiveness of DeamNet is further demonstrated.

4. Conclusion

In this paper, we propose a novel deep network for image denoising. Different from most of the existing deep network-based denoising methods, we incorporate the novel ACP term into the optimization problem, and then the optimization process is exploited to inform the deep network design by using the unfolding strategy. Our ACP-driven denoising network combines some valuable achievements of classic denoising methods and enhances its interpretability to some extent. Experimental results show the leading denoising performance of the proposed network.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Grant 61801316 and the National Postdoctoral Program for Innovative Talents of China under Grant BX201700163.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1692–1700, Jun. 2018. 1, 6, 8

[2] Josue Anaya and Adrian Barbu. Renoirca dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. 6

[3] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3155–3164, Oct. 2019. 1, 2, 7, 8

[4] Antoni Buades, Bartomeu Coll, and J. M. Morel. A non-local algorithm for image denoising. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 60–65, Jun. 2005. 1

[5] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *European Conference on Computer Vision (ECCV)*, pages 171–187, Sep. 2020. 2

[6] Chang Chen, Zhiwei Xiong, Xinmei Tian, and Feng Wu. Deep boosting for image denoising. In *European Conference on Computer Vision (ECCV)*, pages 3–19, Oct. 2018. 1, 2

[7] Fei Chen, Lei Zhang, and Huimin Yu. External patch prior guided internal clustering for image denoising. In *IEEE International Conference on Computer Vision (ICCV)*, pages 603–611, Dec. 2015. 1

[8] Xiaowu Chen, Dongqing Zou, Steven Zhou, Qinping Zhao, and Ping Tan. Image matting with local and nonlocal smooth priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1902–1907, Jun. 2013. 2

[9] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, Jun. 2017. 2, 7, 8

[10] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminancechrominance space. In *IEEE International Conference on Image Processing (ICIP)*, pages I–313–I–316, Oct. 2007. 2, 8

[11] 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, Aug. 2007. 1, 2, 7, 8

[12] Martin David R., Doron Tal Charless, Fowlkes, and Malik Jitendra. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision (ICCV)*, pages 416–423, Jul. 2001. 6

[13] Weisheng Dong, Peiyao Wang, Wotao Yin, and Guangming Shi. Denoising prior driven deep neural network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2305–2318, Oct. 2019. 2

[14] 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. 1, 2

[15] Vadim Fedorov and Coloma Ballester. Affine non-local means image denoising. *IEEE Transactions on Image Processing*, 26(5):2137–2148, May 2017. 1

[16] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, May 2010. 6

[17] Clment Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *European Conference on Computer Vision (ECCV)*, pages 560–577, Oct. 2018. 1, 2

[18] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2511–2520, Oct. 2019. 1, 2

[19] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862–2869, Jun. 2014. 1, 7, 8

[20] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1712–1722, Jun. 2019. 8

[21] Jia Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, Jun. 2015. 6

[22] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Zhang Lei. Focnet: A fractional optimal control network for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6047–6056, June 2019. 1, 2

[23] Jingdong Chen, J. Benesty, Yiteng Huang, and S. Doclo. New insights into the noise reduction wiener filter. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1218–1234, Jul. 2006. 1

[24] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3482–3492, Jun. 2020. 1, 2, 8

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, pages 1–15, May 2015. 6

[26] Claude Knaus and Matthias Zwicker. Progressive image denoising. *IEEE Transactions on Image Processing*, 23(7):3114–3125, Jul. 2014. 1

[27] Marc Lebrun, Miguel Colom, and Jean Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Processing On Line*, pages 5:1–54, 2015. 6, 8

[28] Stamatios Lefkimmiatis. Universal denoising networks : A novel cnn architecture for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3204–3213, Jun. 2018. 2, 7

[29] Xiaoyao Li, Yicong Zhou, Jing Zhang, and Lianhong Wang. Multipatch unbiased distance non-local adaptive means with wavelet shrinkage. *IEEE Transactions on Image Processing*, 29:157–169, 2020. 1

[30] Chih-Hsing Lin, Jia-Shiuan Tsai, and Ching-Te Chiu. Switching bilateral filter with a texture/noise detector for universal noise removal. *IEEE Transactions on Image Processing*, 19(9):2307–2320, Sep. 2010. 1

[31] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S. Huang. Non-local recurrent network for image restoration. In *International Conference on Learning Representations (ICLR)*, pages 1680–1689, Dec. 2018. 1, 2

[32] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 773–782, Jun. 2018. 1

[33] Enming Luo, Stanley H. Chan, and Truong Q. Nguyen. Adaptive image denoising by mixture adaptation. *IEEE Transactions on Image Processing*, 25(10):4489–4503, Oct. 2016. 1, 2

[34] Xiaojiao Mao, Chunhua Shen, and Yubin Yang. Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2802–2810, Dec. 2016. 1, 7, 8

[35] Max Mignotte. A non-local regularization strategy for image deconvolution. *Pattern Recognition Letters*, 29(16):2206–2212, Dec. 2008. 2

[36] Ha Q. Nguyen, Emrah Bostan, and Michael Unser. Learning convex regularizers for optimal bayesian denoising. *IEEE Transactions on Signal Processing*, 66(4):1093–1105, Feb. 2018. 2

[37] 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, Apr. 2017. 1

[38] Jiahao Pang, Gene Cheung, Antonio Ortega, and Oscar C. L Au. Optimal graph laplacian regularization for natural image denoising. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2294–2298, Apr. 2015. 1

[39] Vardan Papyan and Michael Elad. Multi-scale patch-based image restoration. *IEEE Transactions on Image Processing*, 25(1):249–261, Jan. 2016. 1

[40] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1586–1595, Jun. 2017. 6, 8

[41] Tobias Plotz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1087–1098, Dec. 2018. 1, 2, 7, 8

[42] Xie Qi, Zhao Qian, Deyu Meng, Zongben Xu, and Shuhang Gu. Multispectral images denoising by intrinsic tensor sparsity regularization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1692–1700, Jun. 2016. 1

[43] Chao Ren, Xiaohai He, and Truong Q. Nguyen. Adjusted non-local regression and directional smoothness for image restoration. *IEEE Transactions on Multimedia*, 21(3):731–745, Aug. 2019. 2

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, Nov. 2015. 5

[45] Stefan Roth and Michael J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, Feb. 2009. 6

[46] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2774–2781, Jun. 2014. 2

[47] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4549–4557, Oct. 2017. 7, 8

[48] Chen Tao, Kai-Kuang Ma, and Li-Hui Chen. Tri-state median filter for image denoising. *IEEE Transactions on Image Processing*, 8(12):1834–1838, Dec. 1999. 1

[49] Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. Attention-guided cnn for image denoising. *Neural Networks*, 124:117–129, 2020. 7, 8

[50] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020. 7, 8

[51] Radu Timofte, Kyoung Mu Lee, Xintao Wang, Yapeng Tian, Yu Ke, Yulun Zhang, Shixiang Wu, Dong Chao, Lin Liang, and Qiao Yu. Ntire 2017 challenge on single image super-resolution: methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1110–1121, Jun. 2017. 6

[52] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1865–1873, Jun. 2016. 7

[53] Wei Wang, Ruiming Guo, Yapeng Tian, and Wenming Yang. Cfsnet: Toward a controllable feature space for image restoration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4140–4149, Oct. 2019. 7, 8

[54] Ting Xie, Shutao Li, and Bin Sun. Hyperspectral images denoising via nonconvex regularized low-rank and sparse matrix decomposition. *IEEE Transactions on Image Processing*, 29:44–56, 2020. 1

[55] Yuan Xie, Shuhang Gu, Yan Liu, Wangmeng Zuo, Wensheng Zhang, Lei Zhang, and Yuan Xie. Weighted schatten p-norm minimization for image denoising and background subtraction. *IEEE Transactions on Image Processing*, 25(10):4842–4857, Oct. 2016. 1

[56] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *European Conference on Computer Vision (ECCV)*, pages 21–38, Sep. 2018. 1

[57] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1096–1104, Oct. 2017. 1

[58] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and XiangchuFeng. Patch group based nonlocal self-similarity prior learning for image denoising. In *IEEE International Conference on Computer Vision (ICCV)*, pages 244–252, Dec. 2015. 1

[59] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1690–1701, 2019. 2, 8

[60] Kaibing Zhang, Xinbo Gao, Dacheng Tao, and Xuelong Li. Single image super-resolution with non-local means and steering kernel regression. *IEEE Transactions on Image Processing*, 21(11):4544–4556, Nov. 2012. 2

[61] 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, Jul. 2016. 1, 2, 6, 7, 8

[62] Kai Zhang, Wangmeng Zuo, and Zhang Lei. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, Sep. 2018. 2, 7, 8

[63] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision (ECCV)*, pages 294–310, Sep. 2018. 5

[64] Wang Zhou, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. 6