# Supplementary Material: 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

*The following items are contained in the supplementary material:*

*1. Proof of **Theorem 1**.*
*2. Further understanding of DeamNet.*
*3. Analysis of original scheme.*
*4. More results about original scheme and optimized scheme.*
*5. Dual weighting tensors* vs. *single weighting tensor.*
*6. Differences among DeamNet and existing denoising networks.*
*7. Compared with neural architecture search based method.*
*8. Parameter number.*
*9. More qualitative results.*

## A. Proof of Theorem 1

*Proof.* To facilitate the derivation, we introduce some auxiliary variables. Let $\mathcal{Y}=\mathscr{T}(\mathbf{y})\in\mathbb{R}^{n\cdot m}$ be the initial feature tensor of the noisy observation $\mathbf{y}\in\mathbb{R}^n$, $\mathcal{X}=\mathscr{T}(\mathbf{x})\in\mathbb{R}^{n\cdot m}$ be the corresponding feature tensor of the ground-truth image $\mathbf{x}\in\mathbb{R}^n$. Since $\Psi(\mathbf{x}|\mathbf{y},\mathscr{T})=\|\mathcal{Y}-\mathcal{X}\|_2^2$, and $\mathcal{J}^{\star}_{\text{ACP}}(\mathbf{x}|\mathscr{T},\mathscr{K},\mathbf{\Lambda})=\|\mathbf{\Lambda}(\mathcal{X}-\mathscr{K}(\mathcal{X}_k))\|_2^2$, the ACP-driven denoising algorithm can be rewritten as the following optimization problem about $\mathcal{X}$:

$$\hat{\mathcal{X}}=\arg\min_{\boldsymbol{\mathcal{X}}}\|\mathcal{Y}-\mathcal{X}\|_2^2+\lambda\|\mathbf{\Lambda}(\mathcal{X}-\mathscr{K}(\mathcal{X}_k))\|_2^2, \quad (1)$$

where the operators or parameters $\{\mathscr{T}(\cdot),\mathscr{K}(\cdot),\mathbf{\Lambda},\lambda\}$ are preset and $\mathbf{y}$ is the known noisy observation. Therefore, $\{\mathscr{T}(\cdot),\mathscr{K}(\cdot),\mathbf{\Lambda},\lambda,\mathbf{y}\}$ are fixed during the optimization process and $\mathbf{x}$ is the only unknown variable that needs to be estimated. $\mathbf{x}$ can be obtained by applying the reconstruction operator $\mathscr{L}(\cdot)$ to $\hat{\mathcal{X}}$.

Since Eq. (1) is a quadratic optimization problem, we can easily get the closed-form solution via gradient-based method. The gradient of Eq. (1) can be written as follows:

$$\mathcal{L}(\mathcal{X})=2(\mathcal{X}-\mathcal{Y})+2\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}(\mathcal{X}-\mathscr{K}(\mathcal{X}_k)). \quad (2)$$

By constraining the derivative $\mathcal{L}(\mathcal{X})$ of Eq. (1) to 0, we can obtain that

$$(\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda})\mathcal{X}=\mathcal{Y}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}\mathscr{K}(\mathcal{X}_k). \quad (3)$$

Because $\mathbf{\Lambda}=\mathscr{D}(a_1,...,a_l,...,a_{nm})\in\mathbb{R}^{nm\times nm}$ is a diagonal reliability matrix, *i.e.*, $\mathbf{\Lambda}^T=\mathbf{\Lambda}$, we have

$$(\lambda\mathbf{\Lambda}^T\mathbf{\Lambda})^T=\lambda(\mathbf{\Lambda}^T\mathbf{\Lambda})^T=\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}. \quad (4)$$

According to Eq. (4), $\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}$ is also a diagonal matrix. We can easily get that

$$\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}=\mathscr{D}(\lambda a_1^2,...,\lambda a_l^2,...,\lambda a_{nm}^2). \quad (5)$$

Similarly,

$$(\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda})^T=\mathbf{I}^T+(\lambda\mathbf{\Lambda}^T\mathbf{\Lambda})^T=\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}. \quad (6)$$

According to Eq. (6), $\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}$ is also a diagonal matrix. We can easily get that

$$\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}=\mathscr{D}(1+\lambda a_1^2,...,1+\lambda a_l^2,...,1+\lambda a_{nm}^2). \quad (7)$$

Because $(1+\lambda a_l^2)>0$ holds true for arbitrary $l$-s, we have

$$|\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}|=\prod_{l=1}^{nm}(1+\lambda a_l^2)\neq 0, \quad (8)$$

and thus $\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda}$ is an invertible matrix. We can get

$$\begin{aligned}&(\mathbf{I}+\lambda\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}\\ =&(\mathscr{D}(1+\lambda a_1^2,...,1+\lambda a_l^2,...,1+\lambda a_{nm}^2))^{-1}\\ =&\mathscr{D}((1+\lambda a_1^2)^{-1},...,(1+\lambda a_l^2)^{-1},...,(1+\lambda a_{nm}^2)^{-1}).\end{aligned} \quad (9)$$

Thus, we can easily obtain the following pixelwise iterative equation:

$$\begin{aligned}[\mathcal{X}_{k+1},l]&=\frac{[\mathcal{Y},l]+\lambda a_l^2[\mathscr{K}(\mathcal{X}_k),l]}{1+\lambda a_l^2}\\ &=\beta_l[\mathcal{Y},l]+(1-\beta_l)[\mathscr{K}(\mathcal{X}_k),l],\end{aligned} \quad (10)$$

where $[\cdot, l]$ represents the $l$-th element of a tensor, and $\beta_l = 1/(1+\lambda a_l^2) \in (0,\ 1)$. Eq. (10) can be further written into a dual tensor form as

$$\mathcal{X}_{k+1} = \boldsymbol{\beta} \otimes \mathcal{Y} + (\mathbb{1} - \boldsymbol{\beta}) \otimes \mathscr{K}(\mathcal{X}_k) \tag{11}$$

By using $\mathcal{Y} = \mathscr{T}(\mathbf{y})$, $\mathcal{X} = \mathscr{T}(\mathbf{x})$, and $\mathbf{x} = \mathscr{L}(\mathcal{X})$, we can get the final estimate as

$$\mathbf{x}_{k+1} = \mathscr{L}(\boldsymbol{\beta} \otimes \mathscr{T}(\mathbf{y}) + (\mathbb{1} - \boldsymbol{\beta}) \otimes \mathscr{K}(\mathscr{T}(\mathbf{x}_k))). \tag{12}$$

□

## B. Further Understanding of DeamNet

Since our DeamNet is the deep unfolding implementation of the proposed ACP-driven denoising problem, its mathematical explanation has been provided. In this subsection, we will illustrate the effectiveness of DeamNet from the perspective of deep network architecture. 1) The progressive strategy of DeamNet decrease the gap between the estimated clean image and the ground-truth image step-by-step, which reduces the difficulty of image noise removal; 2) the high-dimensional FD module enables DeamNet to transform the original noisy space to a certain FD space, which can better reconstruct high-frequency details. In fact, by regarding the reconstruction error as certain noise, this simple trick can also make the network more useful for other IR applications, *e.g.*, image deblocking; 3) compared with the pixel domain, the high-dimensional FD can also improve the information flow transmission in a deep network, leading to a better fitting ability; 4) in the NLO sub-network, by using the multi-scale strategy, the receptive field can be significantly expanded, and multi-scale features can be obtained for a better feature prediction; 5) to allow the adaptive feature recalibration and across-scale feature interaction for a better network expressive ability, the DEAM module is introduced into the NLO sub-network; 6) the proposed DEAM module also ensures the availability of the low-level information in the long CNN and recalibrates the features in each iteration stage. Therefore, our DeamNet can lead to good denoising performance for both synthetic and real noisy images.

## C. Analysis of Original Scheme

**Convergence of Original Scheme.** In the original scheme, the multi-stage loss function is used to guarantee that the later iteration can generate better features than the layers at previous iterations step-by-step. To analyze the convergence of our original scheme with a multi-stage constraint, we report the PSNR/SSIM results of stage 1, stage 2, stage 3, and stage 4 in Table 1. In addition, the visual results are also provided in Fig. 1. We can find that the results are becoming better with the increase of the stage number on all datasets, which verifies the convergence of the original scheme.

Table 1. Average PSNR (dB) and SSIM values of the reconstructed images by each stage in the original scheme for noise level 25. Set12, BSD68, and Urban100 datasets are tested.

| Dataset | Set12 | BSD68 | Urban100 |
|---|---|---|---|
| Stage 1 | 30.42/0.8628 | 29.17/0.8290 | 29.98/0.8887 |
| Stage 2 | 30.65/0.8679 | 29.32/0.8332 | 30.51/0.8984 |
| Stage 3 | 30.75/0.8700 | 29.37/0.8349 | 30.71/0.9019 |
| Stage 4 | 30.80/0.8713 | 29.39/0.8360 | 30.84/0.9042 |

**Inversion Constraint between FD Module and Reconstruction Module.** According to the derivation, $\mathscr{L}(\cdot)$ should be the inverse operator of $\mathscr{T}(\cdot)$. This can be achieved by adding a branch that only composed by the FD module and the reconstruction module to the main network architecture. In the added branch, the FD module and the reconstruction module share the same parameters as their counterparts in the main network architecture. Moreover, the output of the added branch is forced to be the same as the input of DeamNet. The loss function for the inversion constraint is written as $\frac{1}{N}\sum_{g=1}^{N}\eta\|\mathscr{L}(\mathscr{T}(\mathbf{y}(g))) - \mathbf{y}(g)\|_p^p$. We will show that, by using the loss function for the inversion constraint, the reconstruction module is an approximated inverse operator of the FD module. The experiments are conducted on Set12, BSD68, and Urban100. The PSNR/SSIM results are provided in Table. 2. The visual results of the noisy input image, the feature maps generated by $\mathscr{T}(\cdot)$, and the noisy image projected back to the pixel domain are shown in Fig. 2. The results show that, for the noisy input image, the output by using the cascading FD module and the reconstruction module are very close to the input both objectively and subjectively, which verifies the reversible relationship between $\mathscr{L}(\cdot)$ and $\mathscr{T}(\cdot)$.

To better visualize the feature maps generated by the FD module, the 16 principal feature maps from the original 64 feature maps are provided in Fig. 2. The visual results reveal that the feature maps focus on modelling different frequency components of the noisy image. The first two feature maps reflect the main frequency components of the image, and the noise in these two maps is much lower than that in the noisy input. The other maps responds to different edge/textures of the noisy input image. Consequently, our FD module can extract the hierarchical features from the noisy image and further project them into a high-dimensional space for better dealing with noise removal.

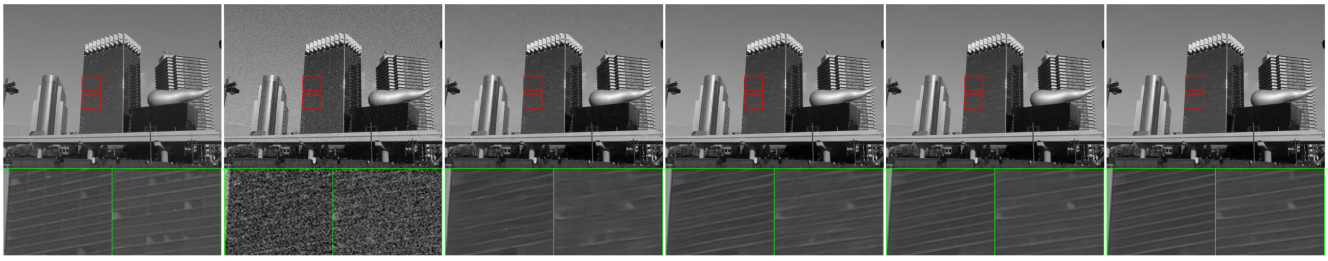


Figure 1. Visual results of the 'Img_086' image from Urban100 in each stage.

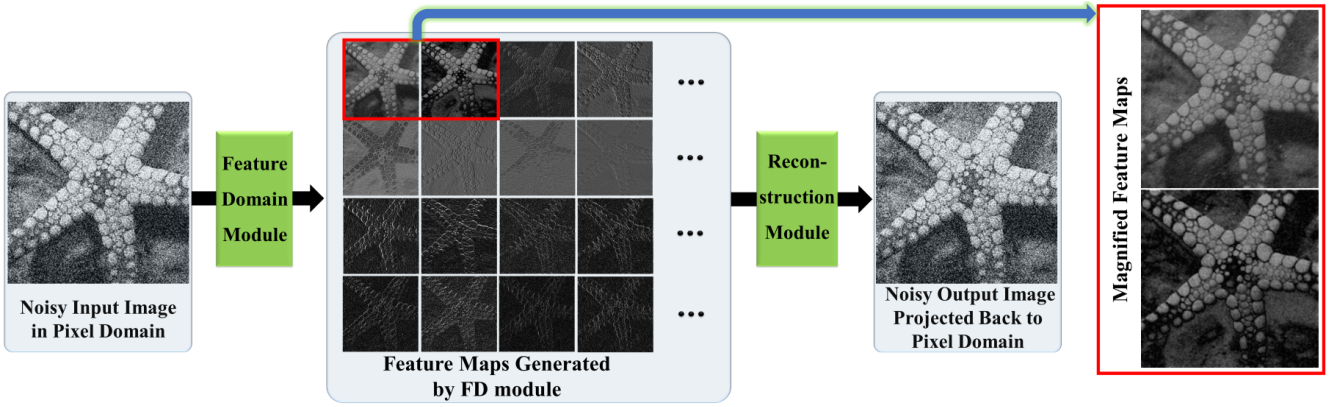


Figure 2. Visual results of the noisy input image, the generated feature maps by $\mathscr{T}(\cdot)$, and the noisy output image projected back to the pixel domain.

Table 2. Average PSNR (dB) and SSIM values of the reconstructed results by the inversion branch. Set12, BSD68, and Urban100 datasets are tested.

| Datasets | Set12 | BSD68 | Urban100 |
|---|---|---|---|
| $\delta = 15$ | 54.57/0.9998 | 54.55/0.9998 | 53.27/0.9997 |
| $\delta = 25$ | 54.39/0.9999 | 54.57/0.9998 | 53.19/0.9999 |
| $\delta = 50$ | 48.25/0.9999 | 48.28/0.9998 | 47.93/0.9998 |

Table 3. Average PSNR (dB) and SSIM values by using the original scheme and the optimized scheme on Set12, BSD68, and Urban100 for noise level 25.

| Scheme | Original Scheme | Optimized Scheme |
|---|---|---|
| Set12 | 30.80/0.8713 | 30.81/0.8717 |
| BSD68 | 29.39/0.8360 | 29.44/0.8373 |
| Urban100 | 30.84/0.9042 | 30.85/0.9048 |

## D. More Results about Original Scheme and Optimized Scheme

The original scheme uses both the inversion constraint and the multi-stage constraint in the loss function of DeamNet. In this subsection, we will provide more results to show that the optimized scheme can achieve slightly better performance than the original version.

The results of using the original scheme and the optimized version are provided in Table 3. The experimental results show that the optimized scheme indeed obtains slightly better performance. This is because the optimized scheme has more freedom in training than the original one, which may lead to better fitting ability. In addition, by constraining $\mathscr{L}(\cdot)$ be the inverse operator of $\mathscr{T}(\cdot)$ in the original scheme may make the network more difficult to train. Therefore, we adopt the optimized scheme as the default scheme in our implementation.

## E. Dual Weighting Tensors vs. Single Weighting Tensor

According to the analysis in our paper, the derived DEAM module can be deemed as a novel attention module. To further prove its effectiveness, we compare the dual version to the non-dual version. Specifically, in the DEAM module, dual weighting tensors ($\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$) are used for the coarse-level feature and the high-level feature, respectively. To evaluate the effect of dual weighting tensors, we remove

Table 4. Average PSNR (dB) and SSIM values of using DEAM and SEAM in DeamNet on Set12, BSD68, and Urban100 for noise level 25.

| DeamNet Variants | with SEAM | with DEAM |
|---|---|---|
| Set12 | 30.73/0.8703 | 30.81/0.8717 |
| BSD68 | 29.40/0.8360 | 29.44/0.8373 |
| Urban100 | 30.75/0.9028 | 30.85/0.9048 |

$\boldsymbol{\alpha}_1$ from DEAM, and only the single weighting tensor $\boldsymbol{\alpha}_2$ is used for the high-level feature (denote it as SEAM, *i.e.*, single element-wise attention mechanism). Table 4 shows that by using dual weighting tensors, higher PSNR/SSIM values can be obtained on all the testing datasets including Set12, BSD68, and Urban100, which verifies the superiority of using the dual weighting tensors in DEAM over the single weighting tensor.

## F. Differences among DeamNet and Existing Denoising Networks

The intuitions of the network design between our DeamNet and the existing networks (*e.g.*, DnCNN [61], FFDNet [62], TNRD [9], RED [34], MemNet [47], UNLNet [28], N$^3$Net [41], FOCNet [22], DPDNN [13], CFSNet [53], ADNet [49], BRDNet [50], RIDNet [3], CBDNet [10], VDN [59], and AINDNet [24], *etc.*) are quite different. Take some networks as examples to illustrate their differences. In our method, a novel image prior, *i.e.*, ACP, is first defined and then exploited to regularize the process of denoising, leading to a model-based method (*i.e.*, the ACP constraint-based denoising method). The network architecture of DeamNet is designed by following the inference process of the proposed model-based method. Among all these previous networks, TNRD [9], FOCNet [22], UNLNet [28], VDN [59], and DPDNN [13], belong to the deep unfolding-based methods. However, the mathematical foundation of DeamNet (*i.e.*, the proposed ACP-driven optimization algorithm) is quite different from these existing networks. For example, TNRD [9] implements the iterative nonlinear reaction diffusion method as a network; the architecture of FOCNet [22] is based on the fractional optimal control theory; UNLNet [28] is based on the non-local variational operator; VDN [59] is based on the variational inference method, which integrates both noise estimation and image denoising into a unique framework; the architecture of DPDNN [13] is derived from the half quadratic splitting method and the plug-and-play framework. In addition, these previous deep unfolding-based networks perform denoising in the pixel domain, while our DeamNet performs denoising in a high-dimensional feature domain. Other deep denosing networks(*e.g.* DnCNN [61], FFDNet [62], RED [34], MemNet [47], CFSNet [53], ADNet [49], BRDNet [50], RIDNet [3], CBDNet [10], and AINDNet [24]) are non-iterative and the model-based methods are not fully considered during their network designs. In contrast, DeamNet employs an iterative strategy similar to the traditional model-based methods for minimizing the gap between the estimated clean image and the ground-truth image step-by-step. Some other works try to integrate some ideas from the traditional methods into the network. For example, N$^3$Net [41] constructs a non-local network by introducing the $k$-nearest neighbor matching operation to the neural network. However, it dose not involve the unfolding operators and only performs denoising in the pixel domain. Overall, our DeamNet is different from these existing methods both in the image prior design and the network architecture design.

Table 5. Denoising performance (PSNR (dB) and SSIM values) on Set12 and BSD68 with the noise level of 50.

| Dataset | CLEARER | DeamNet | CLEARER-P | DeamNet-P |
|---|---|---|---|---|
| Set12 | 27.43/0.8021 | 27.74/0.8057 | 28.08/0.8129 | 28.41/0.8232 |
| BSD68 | 26.31/0.7352 | 26.54/0.7368 | 27.25/0.7681 | 27.55/0.7757 |

## G. Compared with Neural Architecture Search (NAS)-based Method

Recently, NAS methods have attracted much attention and outperformed the existing labor-intensive handcrafted architectures on a few high-level vision tasks. Some methods also applied NAS to image denoising, and achieved promising results. For example, Suganuma *et al.* [1] proposed a convolutional autoencoder designed by the evolutionary algorithm (E-CAE) for image inpainting and denoising. E-CAE showed that simple convolutional autoencoders built upon only standard network components, *i.e.*, convolutional layers and skip connections, can outperform the state-of-the-art methods which employ adversarial training and sophisticated loss functions. Later, Zhang *et al.* [2] proposed HiNAS (Hierarchical NAS), which exploited NAS to automatically design effective neural network architectures for image denoising. More recently, Gou *et al.* [3] presented a novel method termed as multi-sCaLe nEu-

[1] Masanori Suganuma, Mete Ozay, and Takayuki Okatani. Exploiting the potential of standard convolutional autoencoders for image restoration by evolutionary search. *In International Conference on Machine Learning (ICML)*, pages 4771-4780, Jul. 2018.

[2] Haokui Zhang, Ying Li, Hao Chen, and Chunhua Shen. Memory-efficient hierarchical neural architecture search for image denoising, *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3654-3663, Jun. 2020.

[3] Yuanbiao Gou, Boyun Li, Zitao Liu, Songfan Yang, and Xi Peng. CLEARER: Multi-scale neural architecture search for image restoration, *In Advances in Neural Information Processing Systems (NeurIPS)*, pages 1690-1701, Dec. 2020.

ral ARchitecture sEarch for image Restoration (CLEARER), which is a specifically designed NAS network for image restoration. Since CLEARER has reported the denoising results on Set12 and BSD68 (both E-CAE and HiNAS have not reported these results), it is used for a fair comparison. The results for the noise level 50 are listed in Table 5. We can observe that, the results of CLEARER are 27.43dB/0.8021 on Set12 and 26.31dB/0.7352 on BSD68. Ours are 27.74dB/0.8057 and 26.54dB/0.7368, respectively. Gou *et al*. also evaluated CLEARER via patch-wise PSNR/SSIM, denoted by CLEARER-P. Similarly, our DeamNet-P is about 0.31dB/0.0080 higher than CLEARER-P on the image patches. Therefore, the superiority of our DeamNet is verified.

## H. Parameter Number

The computational costs have been provided in the 'Computational Complexity' subsection. In this subsection, we make a comparison of the proposed DeamNet method with other competing approaches on their parameter numbers. Note that 4 stages with shared NLO sub-network parameters are used in DeamNet. The numbers of the parameter *vs*. the average PSNRs of different algorithms are visualized in Figs. 3 and 4. It can be seen from these figures that RED [34], CBDNet [10], VDN [59], and AINDNet(TF) [24] have larger parameter numbers and lower performance when compared with DeamNet. Although other baselines have smaller parameter numbers than ours, their PSNR/SSIM performances are much lower. Consequently, when compared with other state-of-the-art denoising algorithms, the proposed method achieves higher denoising performance with a relatively small parameter number, which demonstrates its effectiveness.

## I. More Qualitative Results

In this subsection, we provide more visual results of different competing approaches to prove the superiority of the proposed DeamNet method over other state-of-the-art image denoising methods. In Figs. 5-7, more results on synthetic noisy images are provided, and in Figs. 8-10, more results on real noisy images are provided. We can observe from these figures that the noise is significantly reduced in the denoised image by DeamNet. Furthermore, image edges and details are reconstructed well. In contrast, the other competing methods may lead to oversmooth results, or generate results with higher remaining image noise than ours. These observations further verify the effectiveness of our DeamNet for synthetic and real-world image denoising both objectively and subjectively.

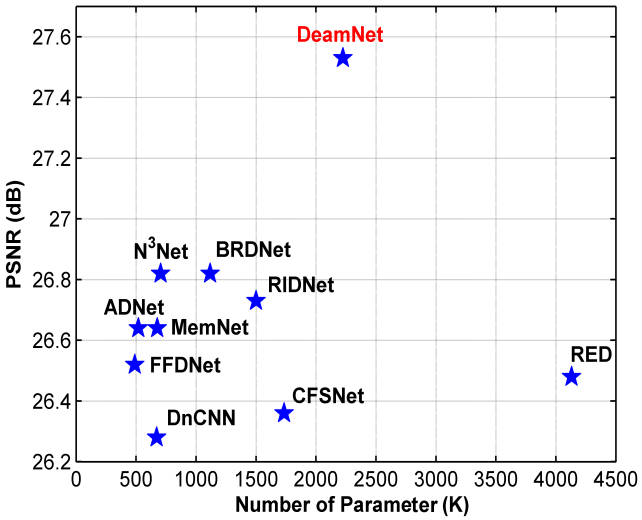


Figure 3. The numbers of parameter and average PSNR values of different models on Urban100 with noise level 50 (synthetic noisy images).

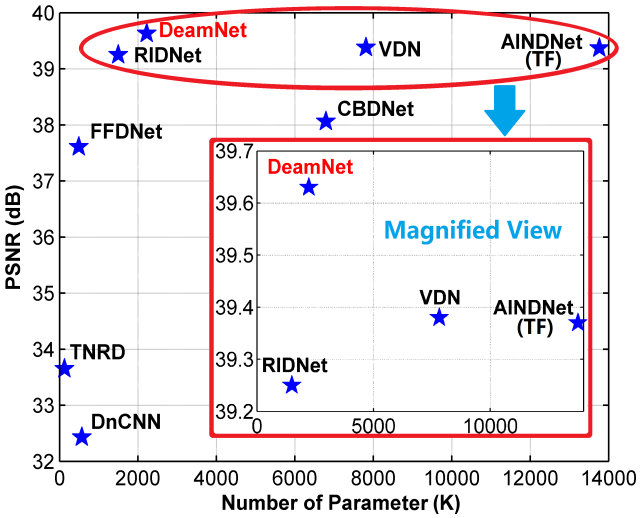


Figure 4. The numbers of parameter and average PSNR values of different models on the DnD benchmark (real noisy images). The magnified view of the ellipsoid region is provided in the rectangle region for better comparison.

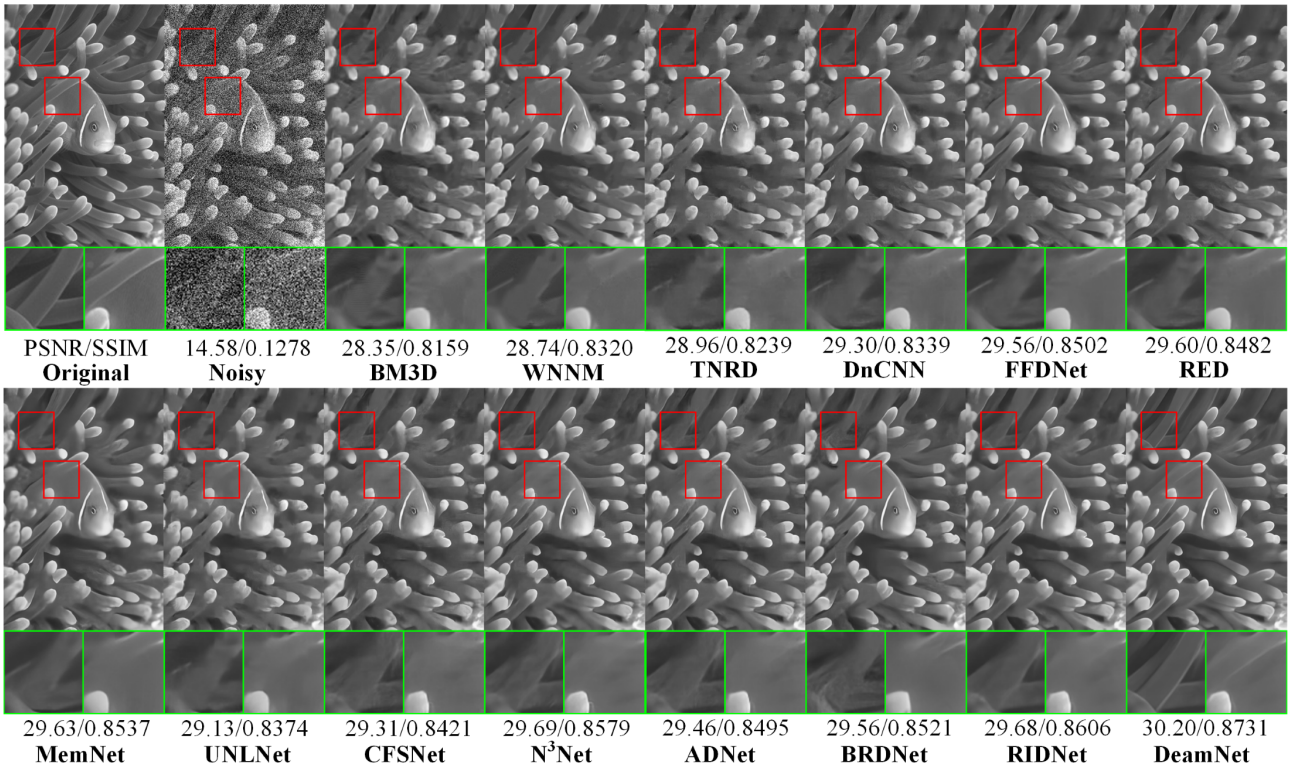


Figure 5. Visual quality comparison for 'test039' from BSD68 (synthetic noisy image).

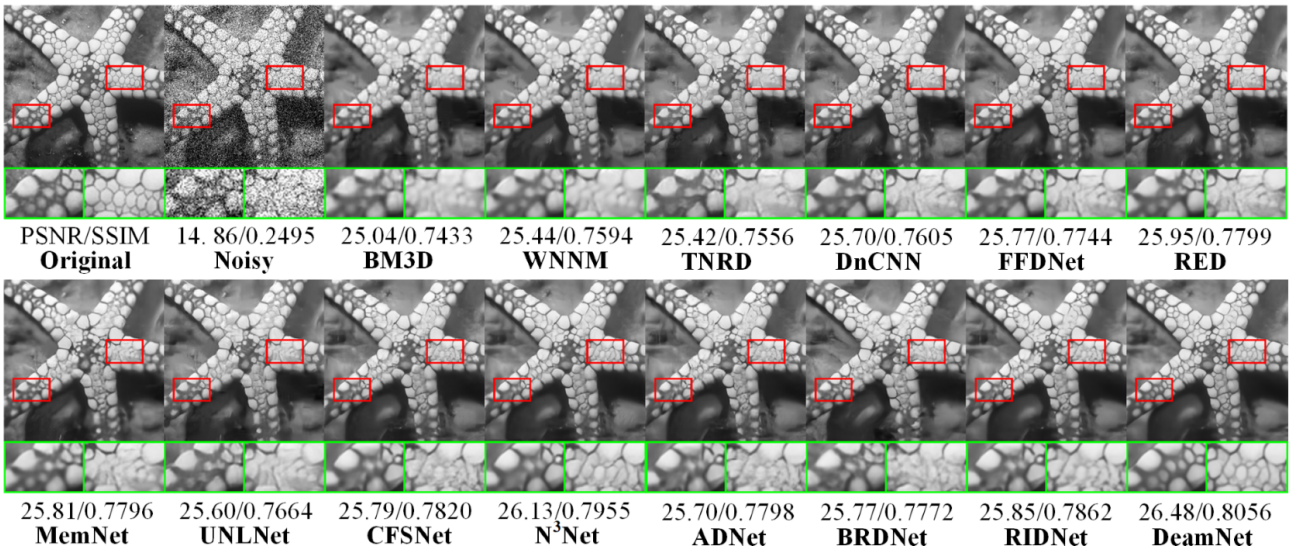


Figure 6. Visual quality comparison for 'Starfish' from Set12 (synthetic noisy image).

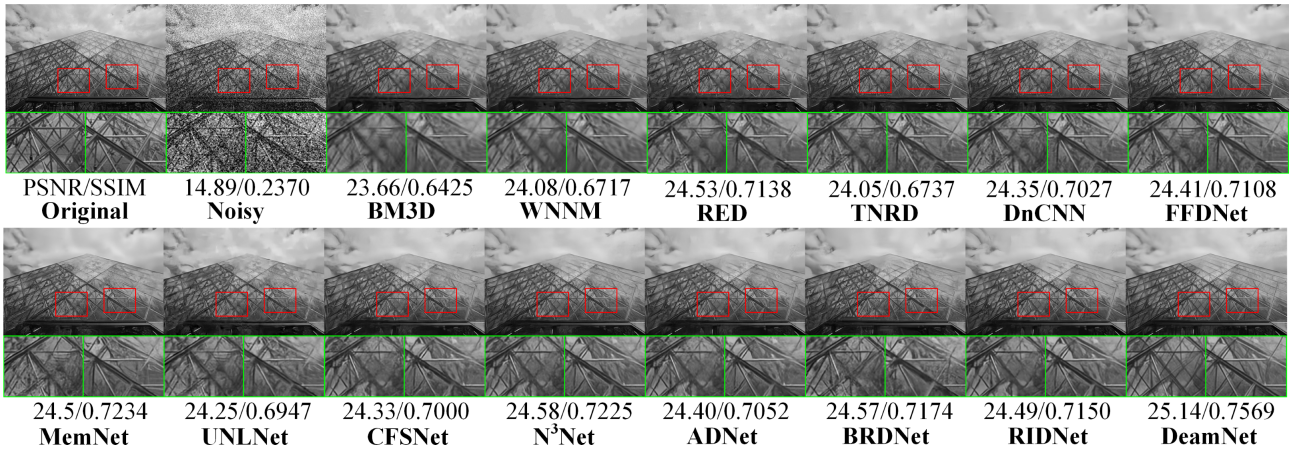


Figure 7. Visual quality comparison for 'test044' from BSD68 (synthetic noisy image).

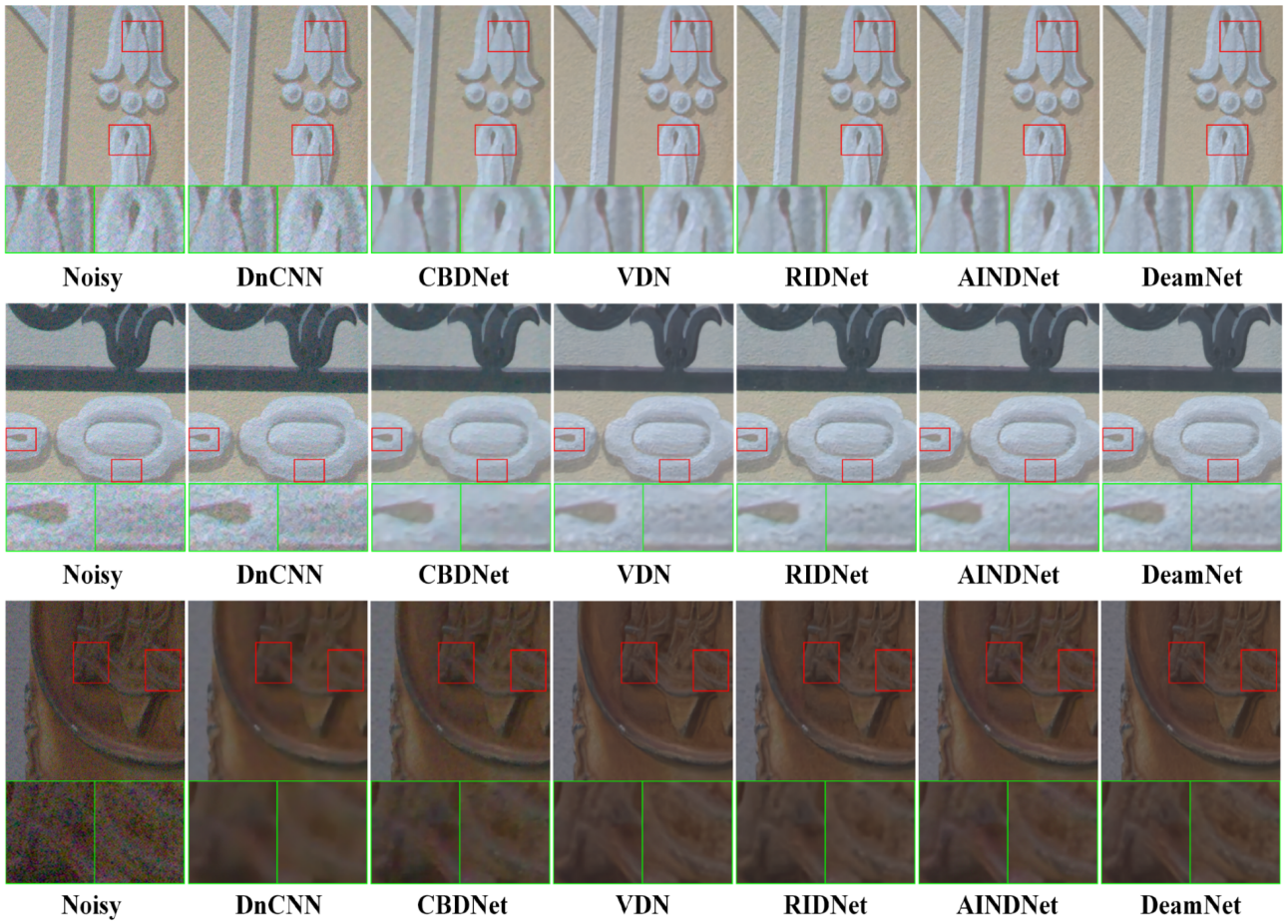


Figure 8. Visual quality comparison for images from DnD (real noisy images).

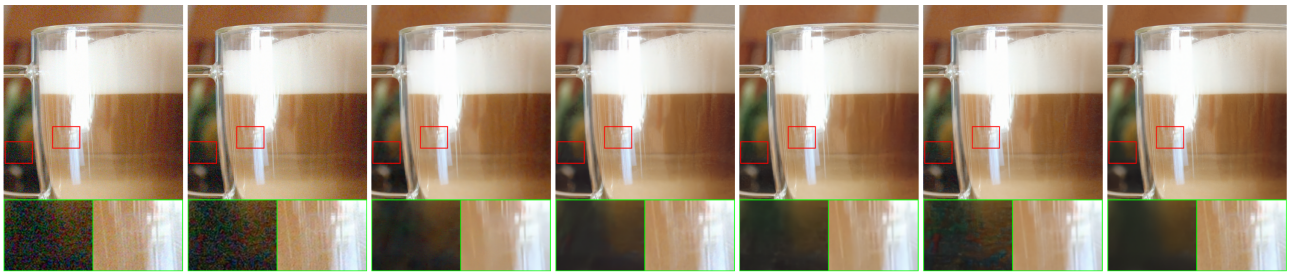


Figure 9. Visual quality comparison for an image from RNI15 (real noisy image).

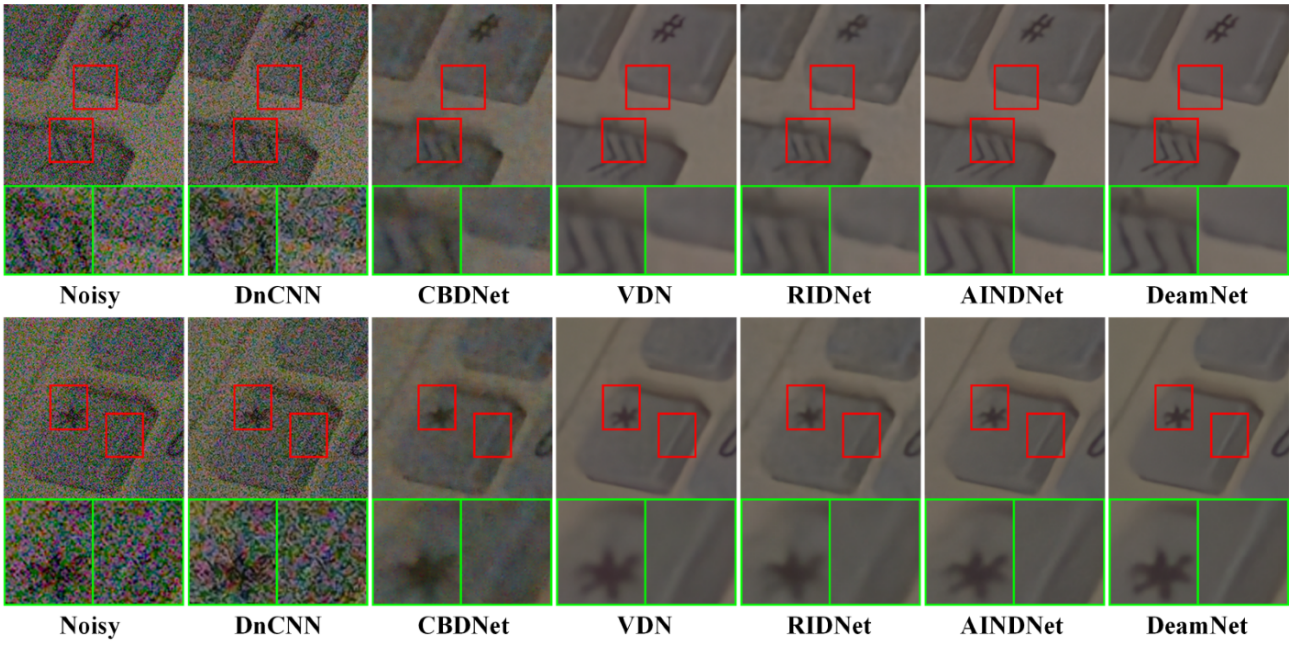


Figure 10. Visual quality comparison for images from SIDD (real noisy images).