Enhanced Latent Space Blind Model for Real Image Denoising via Alternative Optimization

Chao Ren*, Yizhong Pan, Jie Huang

College of Electronics and Information Engineering
Sichuan University, Chengdu, China

Thirty-sixth Conference on Neural Information Processing Systems

Introduction

四川大學 SICHUAN UNIVERSITY

☐ Image denoising methods:

- **Filtering-Based Methods**
  - **Simple.**
  - **Avoiding blurring artifacts** caused by their block-wise operations is **hard**
- **Model-Based Methods**
  - **Strong mathematical derivations.**
  - **No consideration for real noise and external guidance**
- **Learning-Based Methods**
  - ➢ Traditional: Adaptively **learn mapping functions** from datasets. Have **difficulty** in learning an **excellent mapping function**
  - ➢ Deep Network-Based: **Excellent modeling ability**. May be **over-fitted to AWGN** and **fail to handle real noise**

☐ Challenge:

The challenge is incorporating the rich body of **model-based methods** into the design of the **deep network** for removing **real noise** which is much more complicated than AWGN.

The Proposed Method

SICHUAN UNIVERSITY

Enhanced Latent Space Blind Model for Denoising

- **Traditional Model-based Method:** $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathcal{H}(\mathbf{x}, \mathbf{y}) + \mu\varphi(\mathbf{x})$

$\mathbf{x}$: clean image	$\mathbf{y}$: noisy image	$\mathcal{H}(\mathbf{x}, \mathbf{y})$: data fidelity item	$\varphi(\cdot)$: any image regularizer

- **Analysis and Enhancement:**
  - ✓ Performs a constraint in **pixel domain** → Introduce high-dimensional encoding and decoding functions, E and D
  - ✓ **Noise information** is largely ignored → Introduce the real noise map $\mathbf{u}$
  - ✓ **Specific image information** is not exploited effectively for guidance → Introduce certain sophisticated guidance information $\mathbf{g}$ from the image

- **Proposed Method:** $\{\hat{\mathbf{z}}, \hat{\mathbf{u}}\} = \arg\min_{\mathbf{z},\mathbf{u}} \mathcal{H}(\mathbf{z}, \mathbf{u}, \mathbf{y}) + \tau\mathcal{G}(\mathbf{z}, \mathbf{g}) + \eta\psi(\mathbf{u}), s.t., \hat{\mathbf{x}} = D(\hat{\mathbf{z}})$

$\mathcal{G}(\mathbf{z}, \mathbf{g})$: image information guidance term	$\psi(\mathbf{u})$: regularizer for noise information

The Proposed Method

SICHUAN UNIVERSITY

Self-Correction (SC) Alternative Optimization

- According to traditional alternative optimization algorithm, the previous formula can be solved by splitting it into:

z (reconstruction subproblem) **and** **u** (degradation estimation subproblem).

$$\boxed{\mathbf{z}^{(i+1)} = \arg\min_{\mathbf{z}} \mathcal{H}(\mathbf{z}, \mathbf{u}^{(i)}, \mathbf{y}) + \tau \mathcal{G}(\mathbf{z}, \mathbf{g})} \qquad \boxed{\mathbf{u}^{(i+1)} = \arg\min_{\mathbf{u}} \mathcal{H}(\mathbf{z}^{(i+1)}, \mathbf{u}, \mathbf{y}) + \eta \psi(\mathbf{u})} \quad (1)$$

where i is the iteration number. Considering that the last estimates of $\mathbf{z}$ and $\mathbf{u}$ are not used in Eq. (1) and the estimated noise map is available during the iterations, we adopt the SC scheme (introducing $\mathbf{z}^{(i)}$ and $\mathbf{u}^{(i)}$) and specify the form of $\mathbf{g}$.

The Proposed Method

SICHUAN UNIVERSITY

Self-Correction (SC) Alternative Optimization

- Then Eq. (1) becomes:

$$
\begin{cases}
\mathbf{g}^{(i)} = \boxed{G(\mathbf{u}^{(i)})}, \\
\mathbf{z}^{(i+1)} = \arg\min_{\mathbf{z}} \mathcal{H}(\mathbf{z}, \mathbf{u}^{(i)}, \mathbf{y}) + \tau \boxed{\widetilde{\mathcal{G}}(\mathbf{z}, \mathbf{g}^{(i)}, \mathbf{z}^{(i)})} = \boxed{Z(\mathbf{z}^{(i)}, \mathbf{y}, \mathbf{g}^{(i)}, \mathbf{u}^{(i)})}, \\
\mathbf{u}^{(i+1)} = \arg\min_{\mathbf{u}} \mathcal{H}(\mathbf{z}^{(i+1)}, \mathbf{u}, \mathbf{y}) + \eta \boxed{\widetilde{\psi}(\mathbf{u}, \mathbf{u}^{(i)})} = \boxed{U(\mathbf{z}^{(i+1)}, \mathbf{y}, \mathbf{u}^{(i)})},
\end{cases}
\tag{2}
$$

$G(\cdot)$: guidance information generator, $\tilde{\mathcal{G}}(\cdot)$: the joint constraint of GC and SC for $\mathbf{z}$,

$\tilde{\psi}(\cdot)$: the joint constraint of noise information and SC for $\mathbf{u}$,

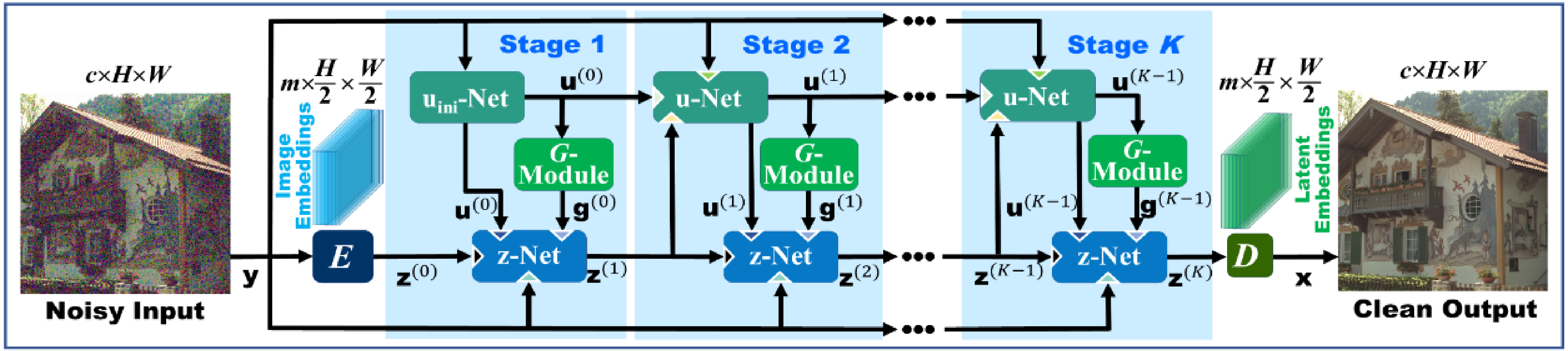
$Z(\cdot)$ and $U(\cdot)$ are the optimization solving processes for the $\mathbf{z}$ and $\mathbf{u}$ subproblems in (2).

Finally, $\mathbf{x}$ can be obtained by applying the LS decoding function D to $\mathbf{z}$.

SCAOED Network

☐ Implicitly Implement SCAOED Algorithm via Deep Network

☞A deep neural network (i.e., ScaoedNet) is constructed by modeling E, D, G, U_{ini}, U, Z.

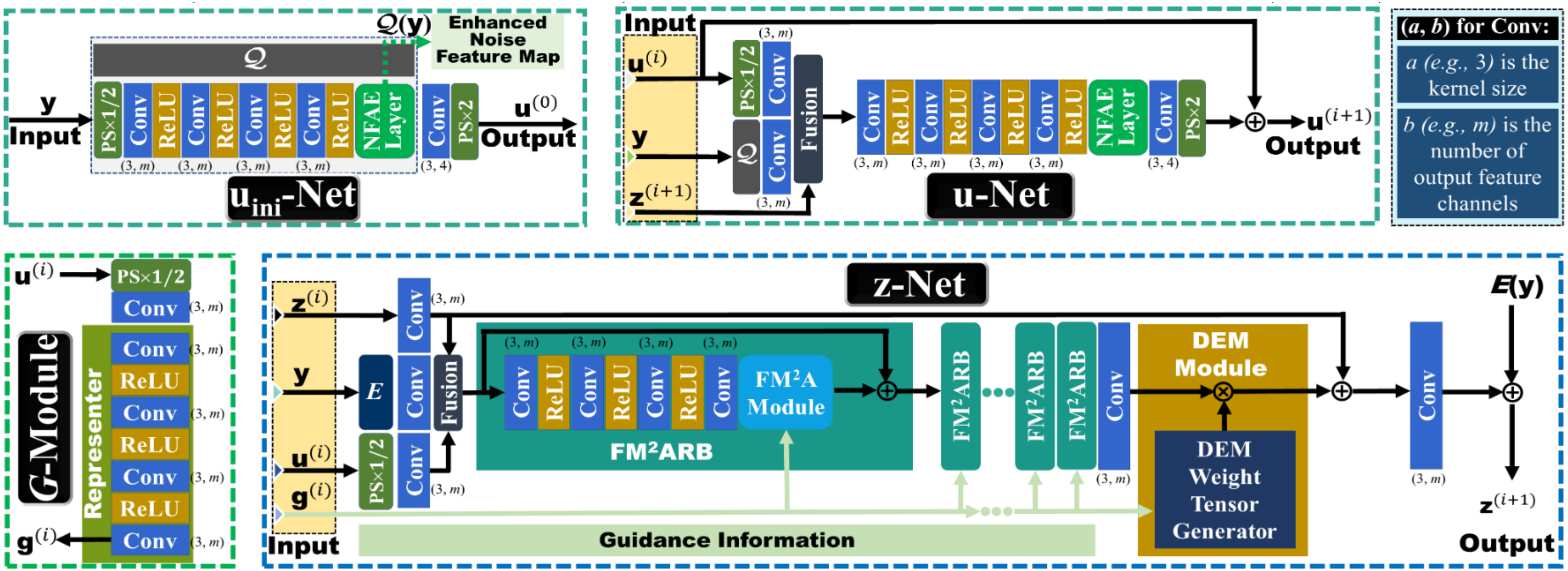


Architecture of the Denoising Network

LS encoding module, LS decoding module, G-module, initial DE network (uini-Net), DE network (u-Net), and RE network (z-Net) are used to implement E, D, G, U_{ini}, U, and Z, respectively.

SCAOED Network

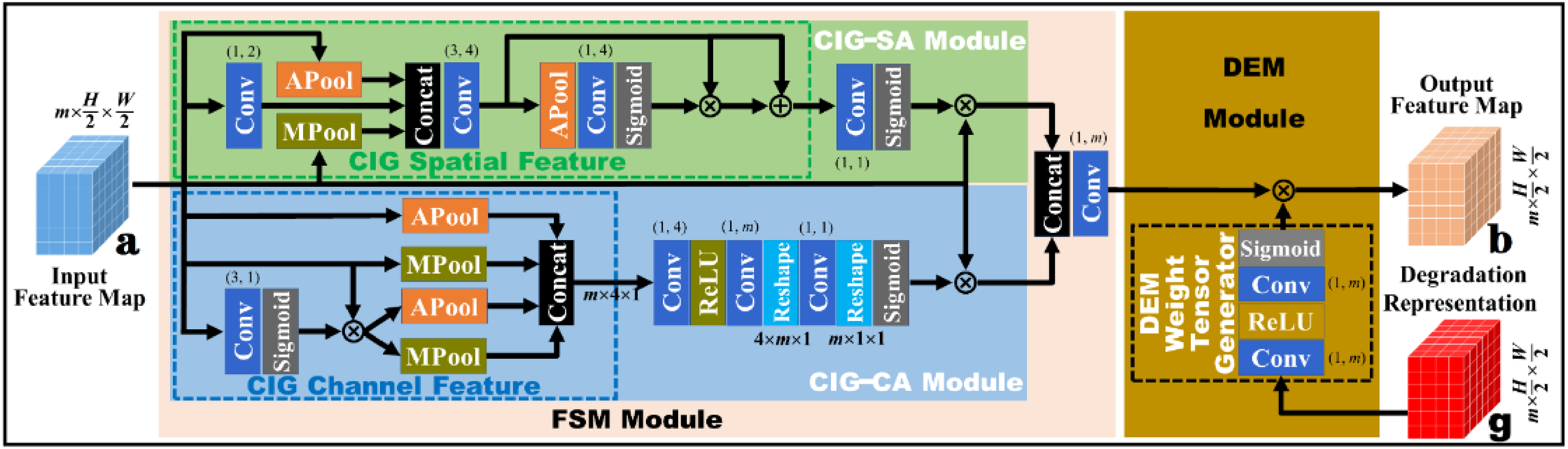
□ Structures of Different Modules



NFAE Layer: $\mathbf{v}(t,h,w) = S\left(10v_{hw}^{t} / (Z_{hw}+\delta),\ 10v_{hw}^{t} / (Z_{t}+\delta)\right)$, $S(a,b) = \text{Sigmoid}(a)\cdot\text{Sigmoid}(b) = \mathbf{1} / \left[\left(\mathbf{1}+e^{-a}\right)\cdot\left(\mathbf{1}+e^{-b}\right)\right]$, $Q(\mathbf{y}) = \mathbf{v} \otimes \mathbf{E}_{in}^{(i+1)}$

SCAOED Network

□ Feature Multiple-Modulation Attention



☞ FSM module consisting of CIG SA and CIG CA, adopts generated spatial and channel weights to recalibrate the input feature map and obtain fused results.

☞DEM module exploits **g** in the implementation of GC to guide the reconstruction of degradation feature map.

Experiments

☐ Implementation Details:

☞ Training Datasets

- Div2K, RENOIR and Smartphone Image Denoising Dataset (SIDD)

☞ Testing Datasets

- SIDD Benchmark, DnD Benchmark and Nam

☞ Suggested Parameter Setting

- Stage number K = 5 and FM2ARB number T = 16.

☞ Experiment Environment

- Ubuntu 16.04, NVIDIA RTX 3090 GPU, Inter(R) Core(TM) i7 5.0GHZ 32G

Experiments

SICHUAN UNIVERSITY

Denoising on Real Noisy Images -- Quantitative Comparison

Table 1: The average PSNR(dB)/SSIM results on DnD benchmark dataset.

Method	Blind/Non-Blind	PSNR↑	SSIM↑
CBM3D[14]	Non-Blind	34.51	0.8507
TNRD[11]	Non-Blind	33.65	0.8306
DnCNN[52]	Blind	32.43	0.7900
FFDNet[53]	Non-Blind	37.61	0.9415
DCDicL[57]	Non-Blind	35.90	0.9150
CBDNet[19]	Blind	38.06	0.9421
VDN[48]	Blind	39.38	0.9518
RIDNet[5]	Blind	39.25	0.9528
AINDNet[22]	Blind	39.37	0.9505
InvDN[27]	Blind	39.57	0.9522
DANet[49]	Blind	39.47	0.9548
DeamNet[36]	Blind	39.63	0.9531
ScaoedNet	Blind	40.12	0.9603
ScaoedNet†	Blind	40.17	0.9597

Table 2: The average PSNR(dB)/SSIM results on SIDD benchmark and validation datasets.

Method	Blind/	SIDD benchmark		SIDD validation	
	Non-Blind	PSNR↑	SSIM↑	PSNR↑	SSIM↑
CBM3D[14]	Non-Blind	25.65	0.685	31.75	0.7061
TNRD[11]	Non-Blind	24.73	0.643	26.99	0.7440
DnCNN[52]	Blind	23.66	0.583	26.20	0.4414
FFDNet[53]	Non-Blind	29.30	0.694	26.21	0.6052
DCDicL[57]	Non-Blind	33.68	0.860	33.76	0.8171
CBDNet[19]	Blind	33.28	0.868	30.83	0.7541
VDN[48]	Blind	39.26	0.955	39.29	0.9109
RIDNet[5]	Blind	37.87	0.943	38.76	0.9132
AINDNet[22]	Blind	38.95	0.952	38.96	0.9123
InvDN[27]	Blind	39.28	0.955	38.30	0.9064
DANet[49]	Blind	39.25	0.955	39.30	0.9164
DeamNet[36]	Blind	39.35	0.955	39.40	0.9169
ScaoedNet	Blind	39.44	0.956	39.48	0.9186
ScaoedNet†	Blind	39.48	0.957	39.52	0.9187

☛ The self-ensemble results denoted by the super script † is also presented to **maximize potential denoising performance** of ScaoedNet.

Experiments

Denoising on Real Noisy Images -- Visual Results

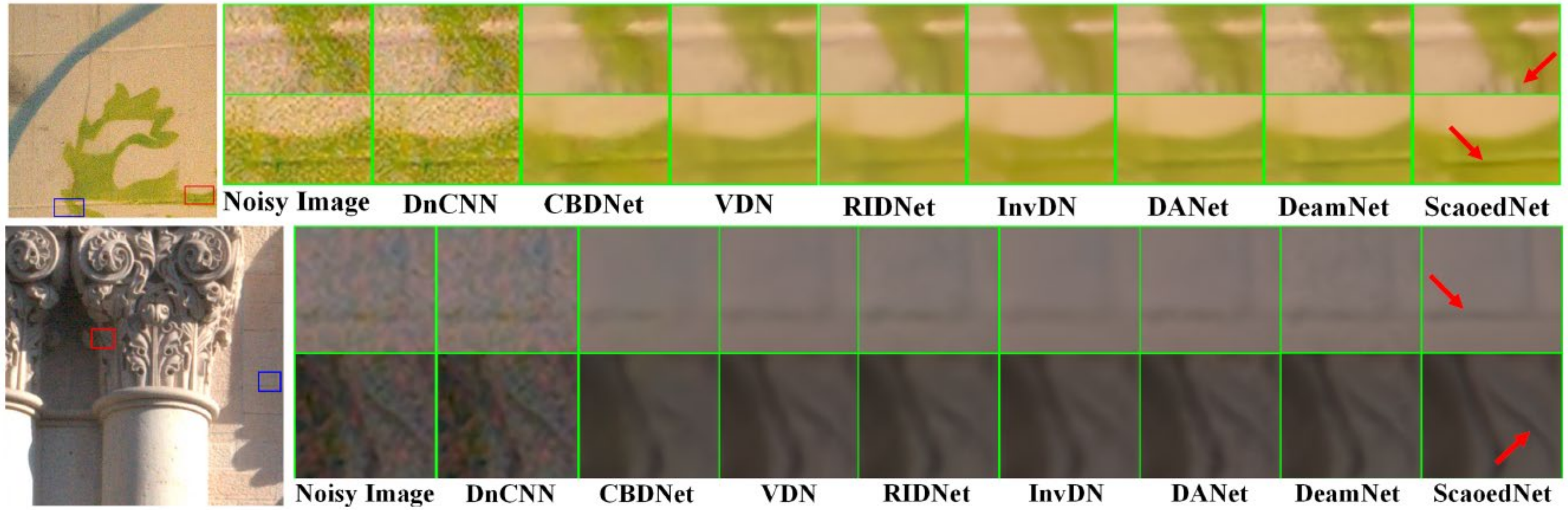


Figure 4
Real denoising results on DnD Benchmark dataset.

☛ ScaoedNet achieves the best visual results.

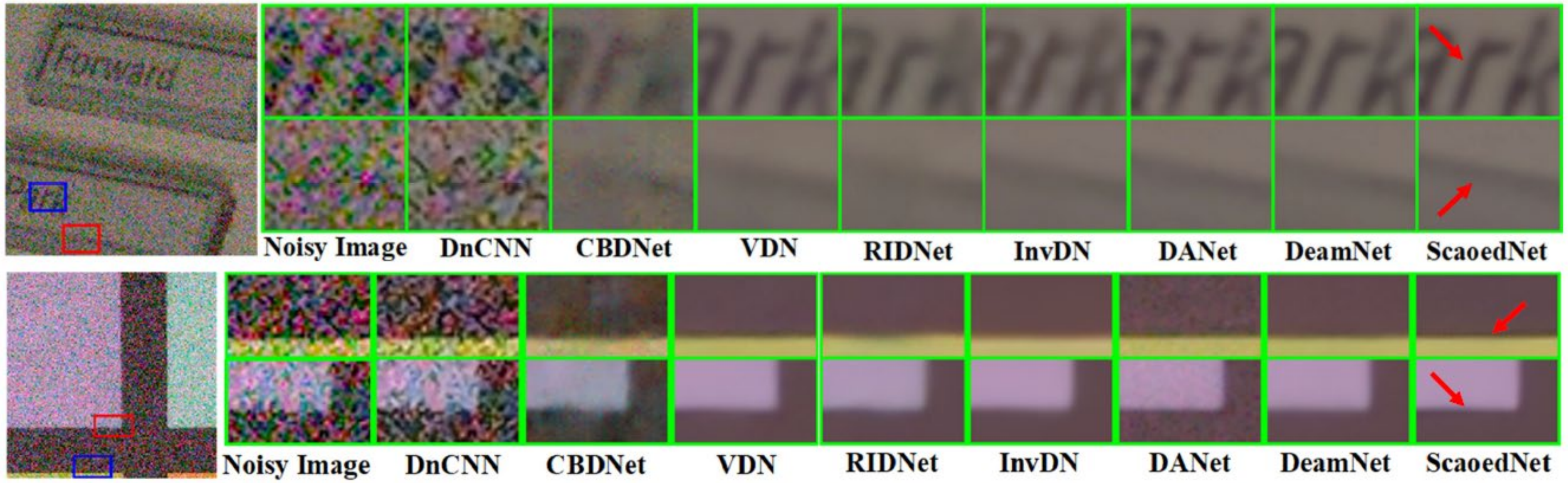


Figure 5
Real denoising results on SIDD validation dataset.

Experiments

Influence of Important Parameters

Table 3 Results on SIDD validation for K-s.

K	1	2	3	4	5	6
PSNR	39.24	39.32	39.40	39.45	39.48	39.48
SSIM	0.9170	0.9177	0.9182	0.9184	0.9186	0.9187

Table 4 Results on SIDD validation for T -s.

T	4	8	12	16	20
PSNR	39.10	39.26	39.40	39.48	39.49
SSIM	0.9158	0.9169	0.9179	0.9186	0.9187

Table S3 L1 distances and PSNRs for verifying the effectiveness of the multiple DE networks.

K	1	2	3	4	5
L_1 distance↓	5.80	5.01	4.48	4.27	4.20
PSNR↑	33.16	34.18	34.80	35.21	35.40

Table S4 Average PSNR (dB) and SSIM values of the denoised images by each stage in ScaoedNet.

Stage	1	2	3	4	5
PSNR↑	36.52	38.96	39.33	39.44	39.48
SSIM↑	0.8729	0.9140	0.9175	0.9184	0.9186

- When stage number K reaches 5, PSNR/SSIM converges.
- With the increasing of DE networks, the estimation result becomes more accurate.
- Considering both the performance and cost, T=16 is the most suitable choice.
- The proposed ScaoedNet is convergent.

Effectiveness of Network Modules

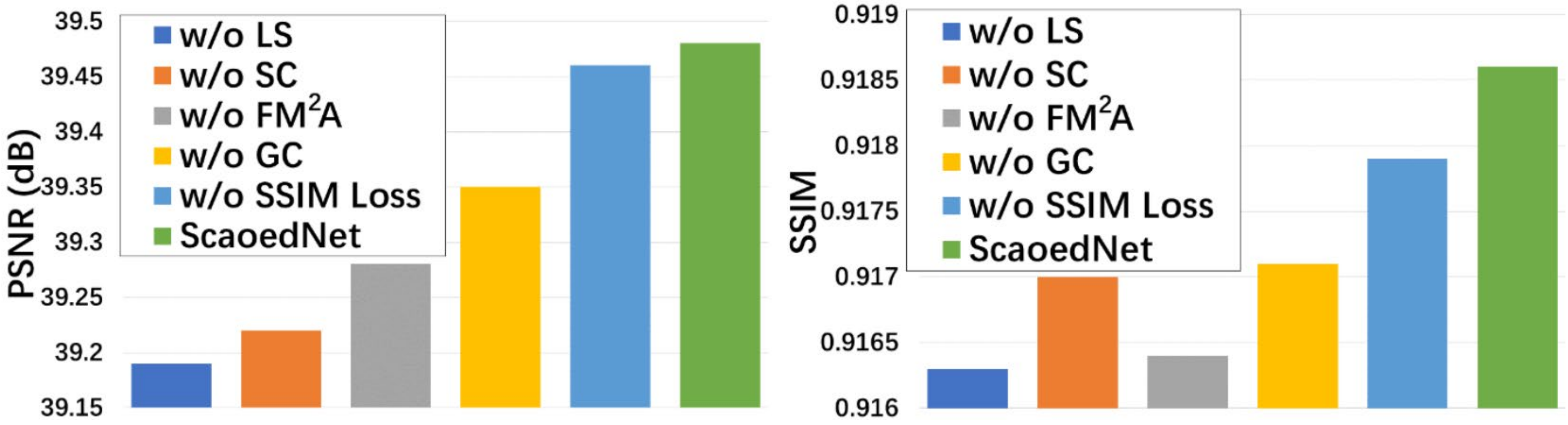


Figure 6 PSNR/SSIM scores of different ScaoedNet variants.

Table 5 The effectiveness of the NFAE layer.

Method	L_1 distance↓	PSNR↑
with NFAE	5.80	33.16
without NFAE	5.97	32.53
CBDNet[19]	6.11	32.30
AINDNet [22]	6.11	32.92
PRIDNet [56]	6.19	33.01

Table S1 PSNRs (dB) and SSIMs on SIDD validation dataset for ScaoedNet with FSM and SCA respectively.

Method	PSNR↑	SSIM↑
ScaoedNet with FSM	39.48	0.9186
ScaoedNet with SCA	39.39	0.9178

Table S2 Ablation on FSM and DEM for SIDD validation dataset.

FSM	✗	✗	✓	✓
DEM	✗	✓	✗	✓
PSNR↑/SSIM↑	39.28/0.9164	39.34/0.9167	39.35/0.9171	39.48/0.9186

Experiments

□ Computational Cost

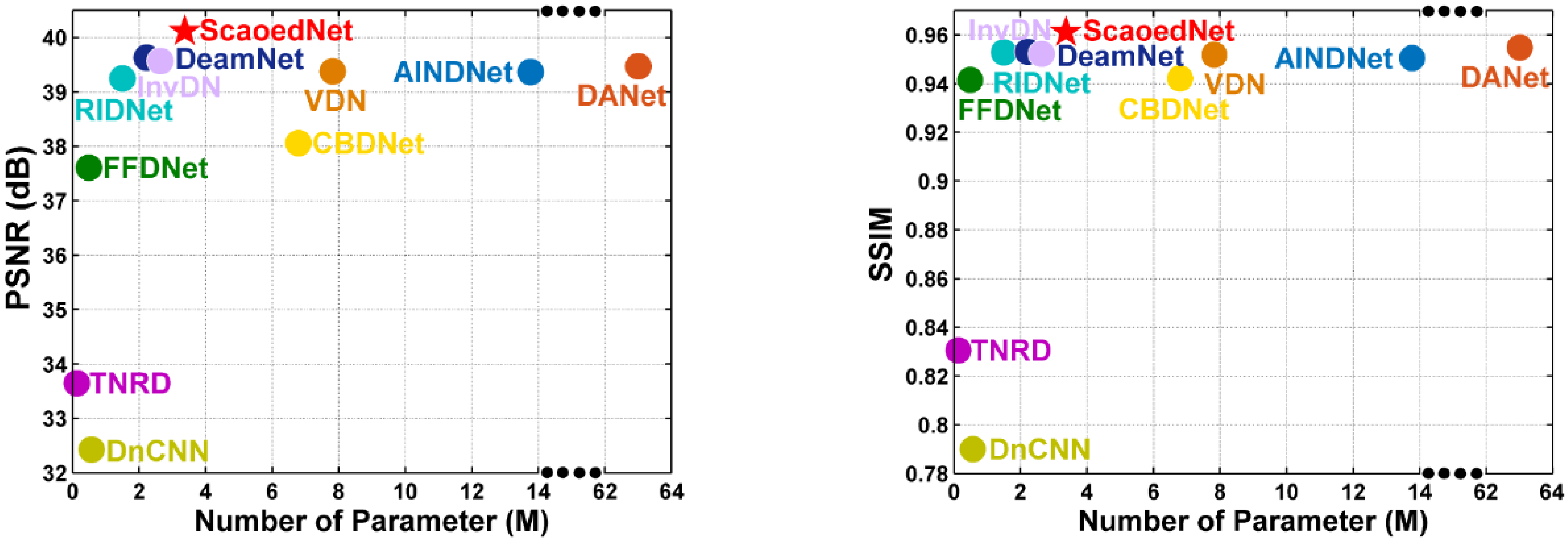


Figure 7 The numbers of parameter vs. average PSNR/SSIM values of different models on DnD.

- Moderate parameter number (about 3M).
- The FLOPS and inference time of ScaoedNet with 1 stage, 3 stages and 5 stages for output resolution 512×512 are 212G/0.07s, 640G/0.15s, and 1071G/0.28s, respectively. For the second-best method DeamNet, it is 589G/0.18s.

Thanks!

NEURAL INFORMATION PROCESSING SYSTEMS

四川大學
SICHUAN UNIVERSITY