CoDe: An Explicit Content Decoupling Framework for Image Restoration

Supplementary Material

We provide more supporting materials in this supplementary file. First, we discuss why the 8×8 block-wise DCT transformation is deprecated. Second, we introduce how to modulate the sub-nets in the IR Network Container for different methods. Then, the effect and effectiveness of each component in our framework are thoroughly ablated, such as the impact of β_i in Content Consistency Loss (CCLoss), the impact of the number of hyperparameter n, and the effectiveness of the divide-and-conquer strategy. Last, more visual comparisons are presented.

A. Why Deprecates the 8×8 DCT?

As shown in Fig. S1, we apply the 8×8 block-wise DCT and its inverse transformation on an arbitrary cropped square image, resulting in the image on the right. It can be seen that this transformation leads to severe visual discontinuity, so we deprecate it and use the typical type-II DCT and iDCT transformation on the whole image instead.

Figure S1. 8×8 block-wise DCT and iDCT transform leads to severe visual discontinuity. Zoom in for more details.

B. How to Modulate the Sub-nets?

The details of modulating the sub-nets of different CNN- and Transformer-based methods can be found as follows: For the CNN-based methods, taking EDSR [7] and $n=2$ as an example, the original EDSR contains a shallow feature extraction layer, 32 duplicated residual blocks with 256 channels of feature layers in each block, and a high-quality feature reconstruction layer following after. $n=2$ means there will be 2 sub-nets, $Subnet_1$ and $Subnet_2$. $Subnet_1$ contains the shallow feature extraction layer and the first 16 residual blocks by modulating the number of channels to 128. $Subnet_2$ contains the remaining 16 residual blocks, whose number of channels still retains the original 256. Then, 2 independent feature reconstruction layers, $Recon_1$ and $Recon_2$, replace the original high-quality feature reconstruction layer and they will transform the corresponding high-dimensional logits to RGB space. For the Transformer-based methods, the construction strategy of different sub-nets is similar to that in EDSR. Taking SwinIR [6] and n=3 as an example, the number of Residual Swin Transformer Blocks (RSTBs) and Swin Transformer Layer (STL) in the original SwinIR are both 6. When setting n to 3, in $Subnet_1$, we retain the shallow feature extraction layer and there are 2 RSTBs, whose number of STL is modulated to 2. As for $Subnet_2$, it cascades another 2 RSTBs, whose number of STL is modulated to 4. $Subnet_3$ consists of the last 2 RSTBs, whose number of STL still keeps 6. After that, $Recon$, containing 3 independent feature reconstruction layers ($Recon_1$, $Recon_2$, and $Recon_3$), will transform the corresponding high-dimensional logits to RGB space respectively. It is worth noting that the smaller the sub-net index number is, the fewer the number of feature layers and channels it contains. Such modulations will bring about a reduction in computation cost, while the computational cost from the extra feature reconstruction layer is trivial. Other networks are reorganized in a similar strategy. We demonstrate the strategy of constructing different sub-nets in the simplest and most direct way, *i.e.*, by modulating the number of channels (width) and feature layers (depth) of the original network to control the computational cost of each sub-net, which indicates our framework can recover the image contents with different patterns well even without any special designs.

C. Ablation Study on Each Component

In this section, we will step-by-step determine the optimal configuration of hyperparameters in each component of our framework and, based on this, delve into the effectiveness of each component.

C.1. How to choose n for each task?

We begin by exploring the influence of the number of decoupled content components under our framework for different tasks. Experiments are first conducted on three typical tasks: single image super-resolution (SISR), grayscale image denoising, and single image motion deblurring. A number of representative methods are selected as the observation objects, and our goal is to determine the corresponding hyperparameter n for each task. Due to the discussion in Section 3.2 in the main manuscript, when n>1, the computational complexity of the IRNC varies with the change of n. Therefore, we need to strike a trade-off between performance and computational cost for each task. The experimental results are shown in Fig. S2. We can find that, for SISR, it achieves a good trade-off between performance and

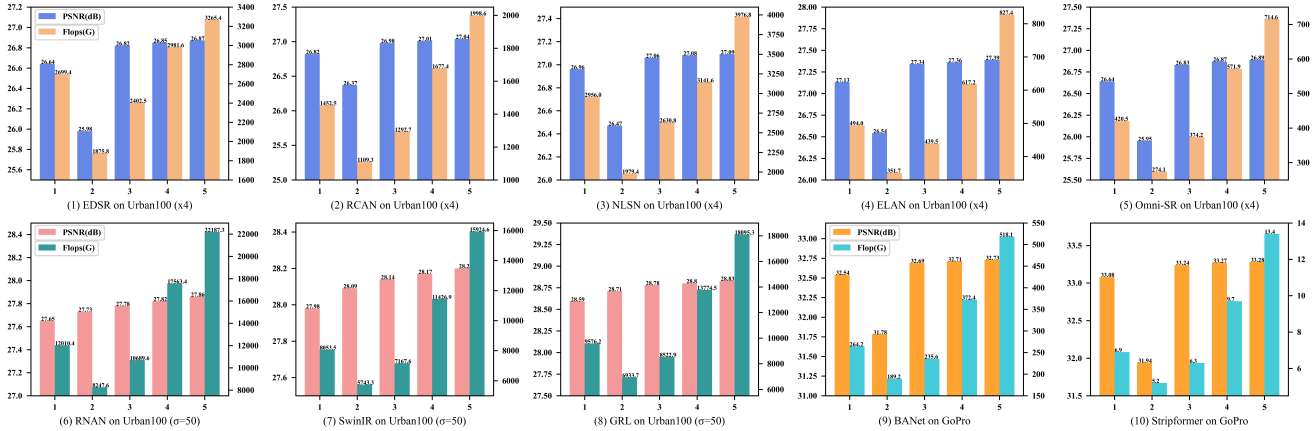

Figure S2. The influence of the number of decoupled content components under our framework for different tasks. The performance (PSNR) and computational complexity (GFlops) are both reported for each method. (1)-(5), (6)-(8), and (9)-(10) denote the representative methods of SISR, grayscale image denoising, and single image motion deblurring respectively. Flops are calculated on average for full-resolution (1280 × 720) images in 4 Nvidia Tesla V100 GPUs.

computation cost when $n=3$. So we set the hyperparameter n to 3 for the SR task, and similarly, we set n to 3 and 2 for image deblurring and denoising respectively. Note that, in this phase, we do not deliberately configure the hyperparameter β_i in the CCLoss and each β_i is simply set to 1.

C.2. Impact of β_i in content consistency loss

Once the hyperparameter n for each task is determined, the number of coefficients in CCLoss is also determined accordingly. We first investigate the impact of configuring different β_i on the performance of each task. For image denoising, image SR, and image deblurring, we set n=2, 3, 3 and retrain our framework on SwinIR [6], EDSR [7], and BANet [11] using Set12 [12] with $\sigma=50$ noise level, Urban100 [4] with $\times 4$ scale, and GoPro [8] for validation, respectively.

The results are shown in Table S1. We can observe that, for image SR, fixing β_1, β_2, and β_3 to 0.3, 0.7, and 1 respectively yields better reconstruction results. Meanwhile, only when allocating comparable weights to the decoupled content components and placing more emphasis on these content with more complex patterns (β_3=1), is there a noticeable improvement in reconstruction performance. These observations indicate: 1) the image SR task focuses on the reconstruction of various content patterns; neglecting any content would adversely affect performance; 2) when performing image SR, the image content corresponding to diverse distribution contributes differently. As for image deblurring, similarly, setting β_1, β_2, and β_3 to 0.3, 0.5, and 1, respectively, achieves better deblurring results. This hints that the deblurring task also tends to focus more on the contents with more complex patterns ($\beta_3 = 1$). For image denoising, setting β_1 and β_2 to 0.3 and 1 respectively can obtain better denoising results. When assigning larger weights to β_1, worse noise removal results are achieved. This demonstrates that it pays more attention to the noised content with more complicated distribution (β_2=1).

Table S1. Effects of different β_i configurations in proposed Content Consistency Loss.

Method	β_1	β_2	β_3	PSNR(dB)	SSIM
SwinIR [6]	1	1	✗	28.09	-
	0.7	1	✗	28.11	-
	0.5	1	✗	28.13	-
	0.3	1	✗	**28.15**	-
	0.1	1	✗	28.14	-
EDSR [7]	1	1	1	26.82	0.8074
	0.7	0.7	1	26.83	0.8076
	0.5	0.7	1	26.85	0.8077
	0.3	0.7	1	**26.89**	**0.8081**
	0.5	0.5	1	26.86	0.8078
	0.3	0.5	1	26.87	0.8079
	0.1	0.5	1	26.84	0.8076
BANet [11]	1	1	1	32.69	0.9572
	0.7	0.7	1	32.72	0.9571
	0.5	0.7	1	32.73	0.9574
	0.3	0.7	1	32.74	0.9577
	0.5	0.5	1	32.72	0.9575
	0.3	0.5	1	**32.76**	**0.9580**
	0.1	0.5	1	32.74	0.9574

Note that the essence of CCLoss is to minimize the DCT coefficients' distance between the reconstructed image and those of the GT image. So, we visualize the DCT coefficients of the LQ image, the reconstructed HQ image with/without the proposed loss function, and the GT image respectively. Taking image SR as an example, we use

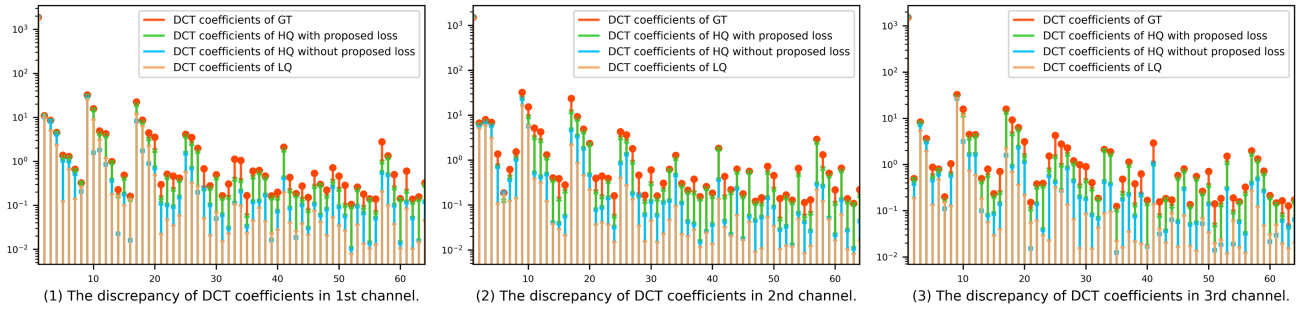

Figure S3. The discrepancy of DCT coefficients of the 3 channels in RGB color space. LQ means the LR image with bicubic upsampling.

GRL [5] as the backbone for $\times 2$ SR, randomly crop 100 8×8 patches at the corresponding positions of these 4 kinds of images, and compute the average DCT coefficients for each image's 3 channels respectively. The result is shown in Fig. S3. It can be observed that under the supervision of the proposed loss function, the DCT coefficients in the reconstructed image are closer to the corresponding DCT coefficients in the GT image than that in the other 2 kinds of images, which demonstrates the effectiveness of the proposed loss function.

C.3. Impact of the number of hyperparameter n

The number of n denotes the number of content components to be decoupled. At the beginning of Section C.1, we first conduct experiments to evaluate the effect of different n in different IR tasks. For further analysis, we combine the denoising results trained on RNAN [14] tested Urban100 [4] with noise level $\sigma = 50$, the super-resolution results trained on EDSR [7] tested on Urban100 [4] with $\times 4$ scale factor, and the deblurring results trained on BANet [11] tested on HIDE [10] to form Table S2. Note that to investigate the impact of the number of hyperparameter n all the β_i in three tasks are set to 1.

We can observe that, for image denoising, as long as the content decoupling operation is performed (n>1), even with much less computational cost compared to the original network, the performance can still be improved. We further conduct the following experiment: setting n to 2, which means only an α_1 is used for content decoupling. Then, we repeat 5 training processes on RNAN [14] while observing the variation trend of α_1. The experimental results are shown in Fig. S4(a). We find that regardless of the random initialization of α_1, it will eventually converge to around 0.9. This indicates that, when conducting noise removal, our framework tends to focus more on the image content with a more complicated distribution that contains the noise. From Table S2, for image SR, we observe a severe performance decrease when setting n=2 compared to the original network (n=1). This is primarily due to the decrease in the network's fitting ability, indicating that the heavy reduction

Table S2. The effects of the number of hyperparameter n.

Method	the number of n	Flops(G)	PSNR(dB)	SSIM
RNAN [14]	1(Original)	12010.4	27.65	-
	2	8247.6	27.73	-
	3	10689.6	27.78	-
	4	17563.4	27.82	-
	5	22187.3	27.86	-
EDSR [7]	1(Original)	2699.4	26.64	0.8033
	2	1875.8	25.98	0.7994
	3	2402.5	26.82	0.8037
	4	2981.6	26.85	0.8038
	5	3265.4	26.87	0.8037
	6	4186.8	26.88	0.8039
	7	5231.5	26.89	0.8038
BANet [11]	1(Original)	264.2	30.16	0.9300
	2	189.2	29.39	0.8531
	3	235.6	30.42	0.9317
	4	372.4	30.44	0.9318
	5	518.1	30.45	0.9320
	6	673.6	30.47	0.9322
	7	859.4	30.46	0.9321

in Flops would hardly make the network catch degraded features more accurately. However, when n=3, our method achieves a good trade-off between performance and computational complexity. As n increases beyond 3, the computational cost also increases due to the reuse of sub-nets with smaller index numbers in the IR Network Container, while the performance gains are marginal. Furthermore, we also set n to 3, which means using α_1 and α_2 for content decoupling, and conduct 5 replicated training processes on EDSR [7] while observing the trend of α_1 and α_2 as well. The results are shown in Fig. S4(b) and (c), where α_1 and α_2 converge to around 0.2 and 0.9, respectively. This indicates that, for image SR, our framework tends to disentangle the image component with diverse content distribution from LR. For image deblurring, we set n to 3 and also conduct 5 replicated training processes on BANet [11] while observing the trend of α_1 and α_2. The results shown in Fig. S4(d) and (e) indicate that image deblurring has similar conclusions to image SR.

It is worth noting that, unlike other frequency decoupling

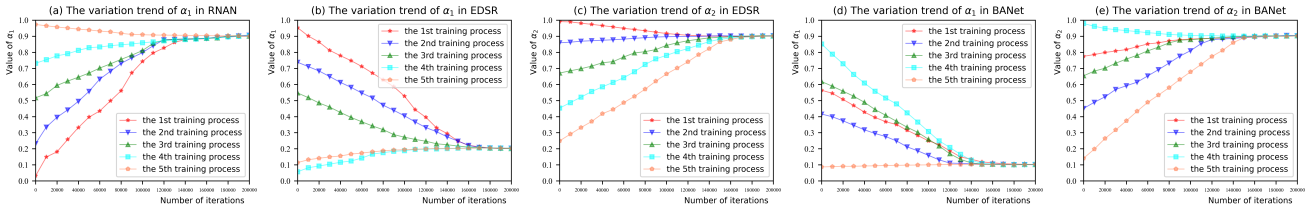

Figure S4. The variation trends of different randomly initalized α_i for 3 tasks. Five repeated training processes are conducted on corresponding methods and the value of each α_i is observed at the first 200,000 iterations.

Table S3. Effects of different content components restored by different sub-nets. C1, C2, and C3 represent the content components **sequentially** decoupled from the Content Decoupling Module respectively. S1, S2, and S3 denote $Subnet_1$, $Subnet_2$, and $Subnet_3$.

RNAN [14] on Set12 [12] with σ=50						ELAN [13] on Set5 [2] with $\times$4 scale factor								GRL [5] on HIDE [10]							
	Case0	Case1	Case2	Case3	Case4		Case0	Case1	Case2	Case3	Case4	Case5	Case6		Case0	Case1	Case2	Case3	Case4	Case5	Case6
-	-	-	-	-	-	C1	✗	S1	S1	S2	S2	S3	S3	C1	✗	S1	S1	S2	S2	S3	S3
C1	✗	S1	S2	S1	S2	C2	✗	S2	S3	S3	S1	S1	S2	C2	✗	S2	S3	S3	S1	S1	S2
C2	✗	S2	S2	S1	S1	C3	✗	S3	S2	S1	S3	S2	S1	C3	✗	S3	S2	S1	S3	S2	S1
PSNR(dB)	27.70	**27.83**	27.59	27.62	27.44	PSNR(dB)	32.75	**32.86**	32.69	32.50	32.57	32.63	32.48	PSNR(dB)	31.65	**31.76**	31.52	31.26	31.67	31.35	31.22
SSIM	-	-	-	-	-	SSIM	0.9022	**0.9031**	0.9014	0.8999	0.9011	0.9013	0.8997	SSIM	0.947	**0.948**	0.933	0.924	0.942	0.919	0.916

methods mentioned in Section 2 in the main manuscript, our framework allows for more fine-grained content decoupling by predefining the hyperparameter n, resulting in better restoration performance. To obtain good trade-offs between performance and computational cost, we finally set $n=2,3,3$ in both real and synthetic scenes for image denoising, image SR, and image deblurring respectively.

C.4. Effectiveness of divide-and-conquer strategy

We further investigate the effectiveness of the divide-and-conquer strategy on performance by exploring the impact of different $Subnet_i$ handling different content components. The experimental results are shown in Table S3. Specifically, we take image denoising on RNAN [14] as an example. The experiments are divided into 5 cases: Case0 represents the original performance; Case1-4 represent the different content components decoupled by the Content Decoupling Module are alternately fed into $Subnet_1$ and $Subnet_2$. The performance is improved only when the content components sequentially generated from the Content Decoupling Module, *i.e.* C1 and C2, are recovered using $Subnet_1$ and $Subnet_2$, respectively (Case1). As for Case2, when the higher computational complexity $Subnet_2$ is used to handle the content component with a much less complicated distribution (C1), it leads to a certain degree of overfitting and adversely affects performance. The experimental results demonstrate the effectiveness of the proposed divide-and-conquer strategy. Another two tasks get similar conclusions from the experimental results.

D. Additional Visual Results

In this supplementary material, we give more visual comparison results of our CoDe and other most current state-of-the-art methods as the supplement of the visualization in the main manuscript.

Real-world image super-resolution: Fig. S5
Gaussian grayscale image denoising: Fig. S6
Gaussian color image denoising: Fig. S7
Real-world image denoising: Fig. S8 and Fig. S9

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700, 2018. 5

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 4

[3] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3086–3095, 2019. 5

[4] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 2, 3

[5] Yawei Li, Yuchen Fan, Xiaoyu Xiang, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. *arXiv preprint arXiv:2303.00748*, 2023. 3, 4

Figure S5. Visual comparisons of **real-world** image SR. The image is from RealSR [3]. Best viewed by zooming.

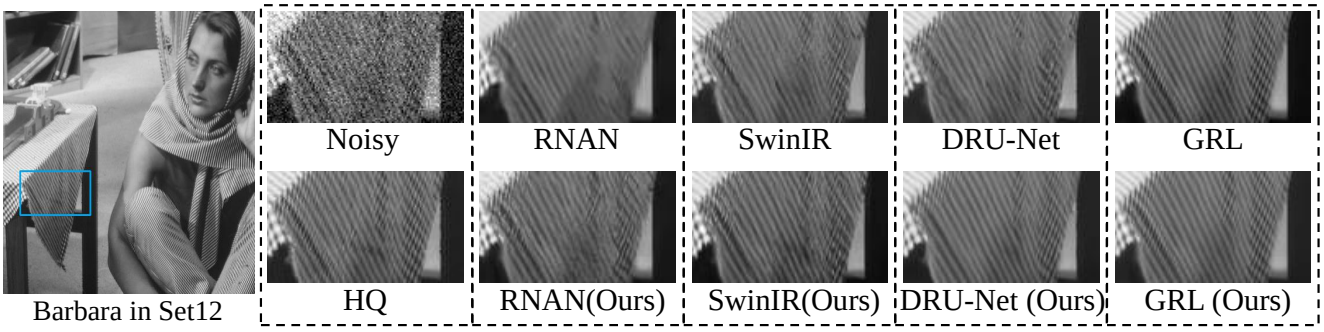

Figure S6. Visual comparison of gaussian **grayscale** image denoising with noise level σ=50. Zoom in for more details.

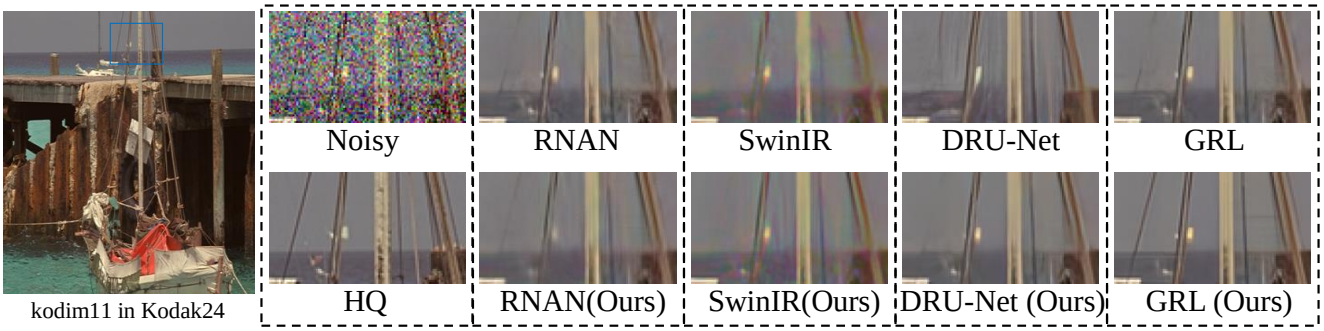

Figure S7. Visual comparison of gaussian **color** image denoising with noise level σ=50. Zoom in for more details.

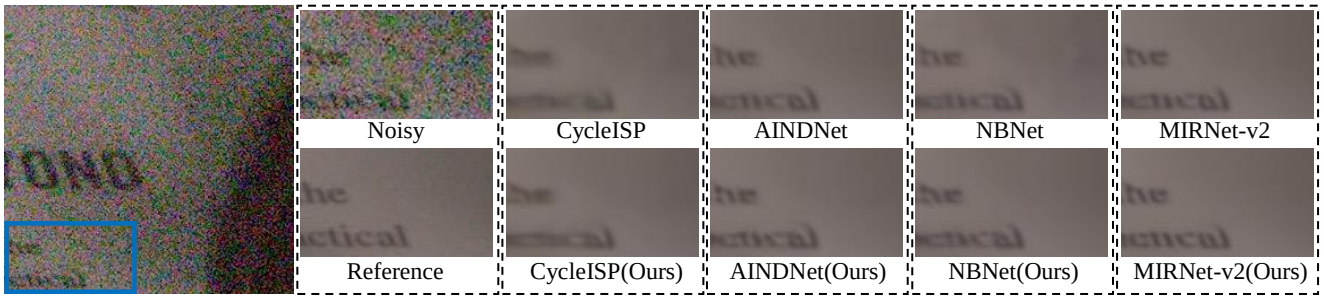

Figure S8. Visual comparisons of the **real-world** image denoising examples from SIDD [1] dataset. Zoom in for a better view.

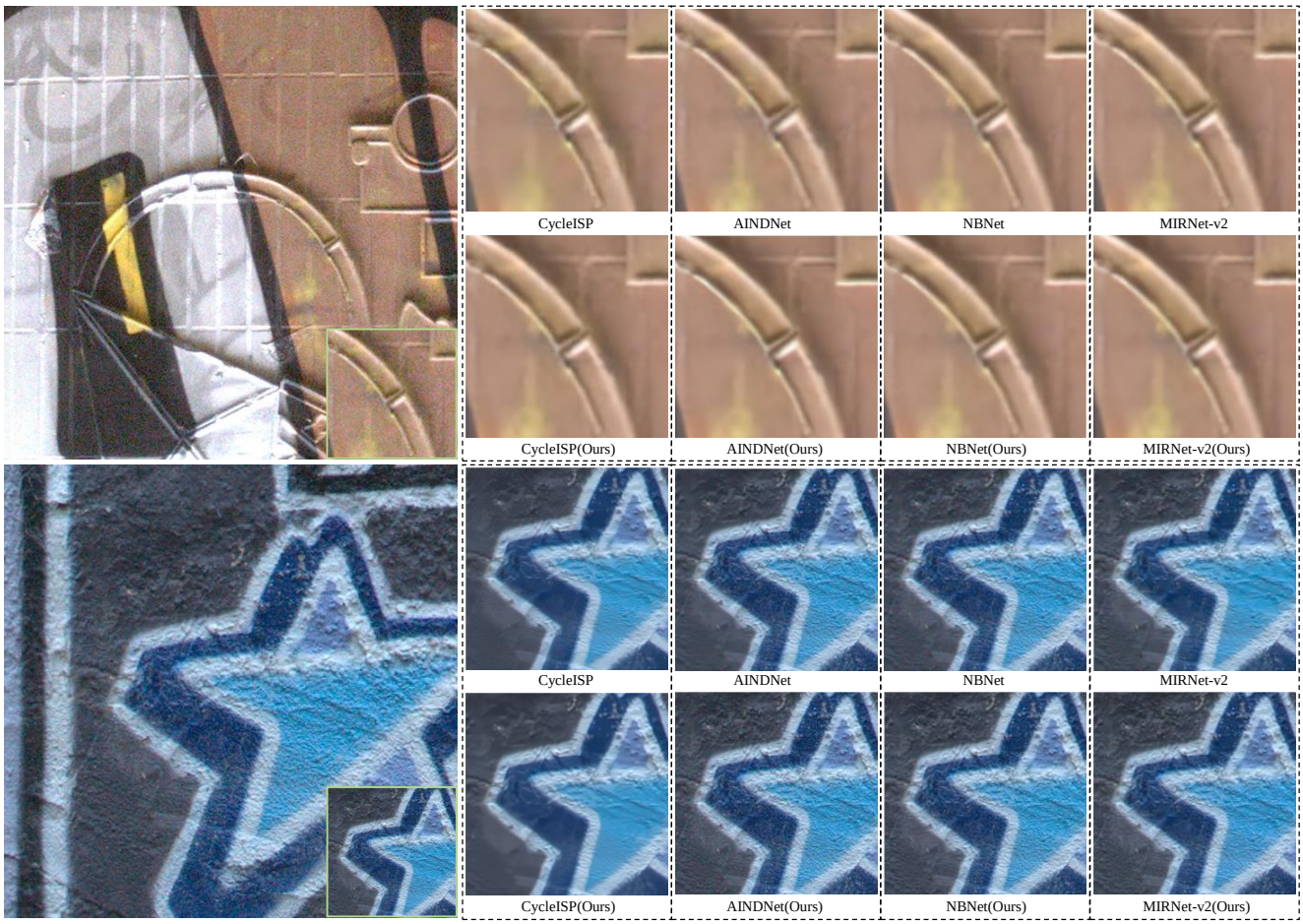

Figure S9. Visual comparisons of the **real-world** image denoising examples from DND [9] dataset. Zoom in for a better view.

[6] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1, 2

[7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2, 3

[8] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 2

[9] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017. 6

[10] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5572–5581, 2019. 3, 4

[11] Fu-Jen Tsai, Yan-Tsung Peng, Chung-Chi Tsai, Yen-Yu Lin, and Chia-Wen Lin. Banet: A blur-aware attention network for dynamic scene deblurring. *IEEE Transactions on Image Processing*, 31:6789–6799, 2022. 2, 3

[12] 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, 2017. 2, 4

[13] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 649–667. Springer, 2022. 4

[14] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019. 3, 4