Noisy-As-Clean: Learning Self-Supervised Denoising From Corrupted Image

Jun Xu, *Member, IEEE*, Yuan Huang, Ming-Ming Cheng, *Senior Member, IEEE*, Li Liu, *Senior Member, IEEE*, Fan Zhu, Zhou Xu, and Ling Shao, *Senior Member, IEEE*

***Abstract*— Supervised deep networks have achieved promising performance on image denoising, by learning image priors and noise statistics on plenty pairs of noisy and clean images. Unsupervised denoising networks are trained with only noisy images. However, for an unseen corrupted image, both supervised and unsupervised networks ignore either its particular image prior, the noise statistics, or both. That is, the networks learned from external images inherently suffer from a domain gap problem: the image priors and noise statistics are very different between the training and test images. This problem becomes more clear when dealing with the signal dependent realistic noise. To circumvent this problem, in this work, we propose a novel "Noisy-As-Clean" (NAC) strategy of training self-supervised denoising networks. Specifically, the corrupted test image is directly taken as the "clean" target, while the inputs are synthetic images consisted of this corrupted image and a second yet similar corruption. A simple but useful observation on our NAC is: *as long as the noise is weak, it is feasible to learn a self-supervised network only with the corrupted image, approximating the optimal parameters of a supervised network learned with pairs of noisy and clean images.* Experiments on synthetic and realistic noise removal demonstrate that, the DnCNN and ResNet networks trained with our self-supervised NAC strategy achieve comparable or better performance than the original ones and previous supervised/unsupervised/self-supervised networks. The code is publicly available at https://github.com/csjunxu/Noisy-As-Clean.**

***Index Terms*— Image denoising, self-supervision, convolutional neural network.**

Manuscript received May 9, 2020; revised August 21, 2020; accepted September 21, 2020. Date of publication September 30, 2020; date of current version October 6, 2020. This work was supported in part by the Major Project for New Generation of AI under Grant 2018AAA0100400, in part by the Fundamental Research Funds for the Central Universities, Nankai University, under Grant 63201168 and Grant 92022104, in part by the NSFC under Grant 61922046, and in part by the Tianjin Natural Science Foundation under Grant 18ZXZNGX00110. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nikolaos Mitianoudis. *(Jun Xu and Yuan Huang contributed equally to this work.) (Corresponding author: Ming-Ming Cheng.)*

Jun Xu and Ming-Ming Cheng are with the TKLNDST, College of Computer Science, Nankai University, Tianjin 300071, China (e-mail: csjunxu@nankai.edu.cn; cmm@nankai.edu.cn).

Yuan Huang is with the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710054, China.

Li Liu, Fan Zhu, and Ling Shao are with the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates, and also with the Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates.

Zhou Xu is with the School of Big Data and Software Engineering, Chongqing University, Chongqing 400044, China.

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

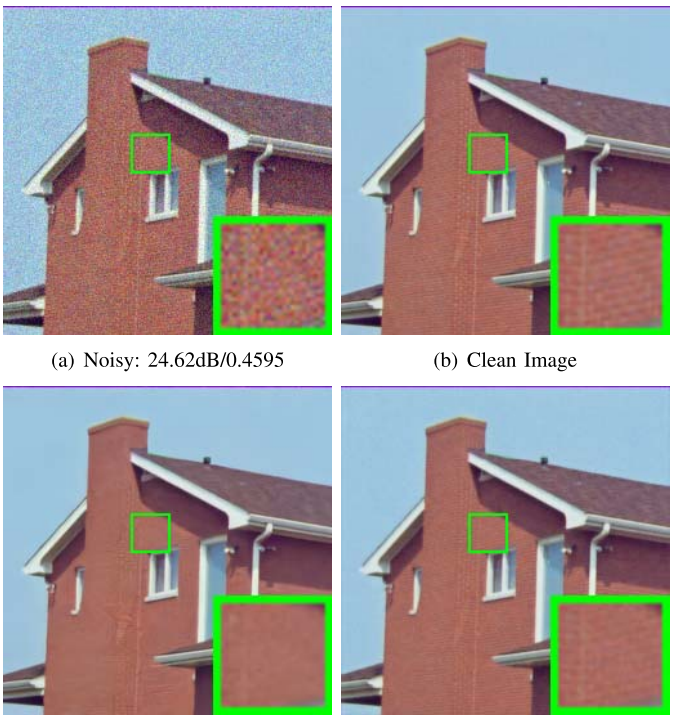
Digital Object Identifier 10.1109/TIP.2020.3026622

I. Introduction

IMAGE denoising is an ill-posed inverse problem to recover a *clean* image $\mathbf{x}$ from the *observed* noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$, where $\mathbf{n}_o$ is the *observed* corrupted noise. One popular assumption on $\mathbf{n}$ is the additive white Gaussian noise (AWGN) with standard deviation (std) σ, which serves as a perfect test bed for supervised networks in the deep learning era [15], [31], [32]. Supervised networks [21], [30], [41] learn the image priors and noise statistics on plenty pairs of clean and corrupted images, and achieve promising denoising performance on the images with similar priors and noise statistics (e.g., AWGN).

With advances on AWGN noise removal [6], [30], [41], a natural question arises is how these denoising networks can exert their effect on real noisy photographs. Realistic noise is signal dependent and more complex than AWGN [28], [29], [35]. Thus, previous supervised denoising networks unavoidably suffer from a *domain gap problem*: both the image priors and noise statistics in training are different from those of the real-world test images. Recently, several unsupervised [7], [19], [22], [23] and self-supervised [3], [20] networks have been developed to get rid of the dependence on clean images, which are difficult to be obtained in real-world scenarios. However, unsupervised networks are subjected to the gap on either image priors or noise statistics, while self-supervised suffer from the gap on noise statistics, between the external images for training and the corrupted ones for test. Besides, several networks [22], [23] succeed on the zero-mean noise. But the realistic noise in real-world images is not necessarily zero-mean [1], [28], [29].

To alleviate the domain gap on image priors and noise statistics between training and test images, in this paper, we propose a "Noisy-As-Clean" (NAC) strategy for training self-supervised denoising networks. In our NAC, we directly train an image-specific network by taking the corrupted image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ as the "*clean*" target. Thus, the domain gap on image priors are largely bridged by our NAC. To reduce the gap on noise statistics, for the target corrupted image $\mathbf{y}$, we take as the input of our NAC a *simulated* noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$ consisting of the corrupted image $\mathbf{y}$ and a *simulated* noise $\mathbf{n}_s$, which is statistically close to the corrupted noise $\mathbf{n}_o$ in $\mathbf{y}$. By this way, our NAC network learns to clean up the *simulated* noise $\mathbf{n}_s$ from the doubly corrupted image $\mathbf{z}$ during



(a) Noisy: 24.62dB/0.4595 (b) Clean Image

(c) DnCNN [41]: 34.23dB/0.8695 (d) DnCNN+NAC: **35.80**dB/**0.9116**

Fig. 1. Denoised images and PSNR/SSIM results of DnCNN [41] (c) and DnCNN trained by our NAC strategy ("DnCNN+NAC") (d) on the color image *House* (b) corrupted by AWGN noise ($\sigma = 15$) (a).

training, and thus is able to remove the noise $\mathbf{n}_o$ from the corrupted image $\mathbf{y}$ during test.

A simple but useful observation about our NAC strategy is: *as long as the corrupted noise is "weak", it is feasible to train a self-supervised denoising network only with the corrupted test image, and the optimal parameters are very close to those of a supervised network trained with a pair of the corrupted image and its clean version.* Though being very simple, our NAC strategy is very effective for image denoising. In Figure 1, we compare the denoised images by the vanilla DnCNN [41] and the DnCNN trained with our NAC (DnCNN+NAC), on the image "House" corrupted by AWGN ($\sigma = 15$). We observe that the "DnCNN+NAC" achieves better visual quality and higher PSNR/SSIM results than DnCNN [41], which is trained on plenty of noisy and clean image pairs. Experiments on diverse benchmarks demonstrate that, when trained with our NAC strategy, the DnCNN [41] and ResNet [15] in Deep Image Prior (DIP) [23] achieve comparable or better performance than supervised denoising networks on synthetic and real-world noisy images. Our work reveals that, *when the noise is "weak"*, a self-supervised network trained directly on the corrupted image can obtain comparable or even better performance than supervised networks on image denoising.

In summary, our contributions are mainly three-fold:

- We propose a "Noisy-As-Clean" (NAC) strategy for training self-supervised denoising networks.
- We provide a theoretical background of our NAC strategy, and implement the DnCNN [41] and ResNet in DIP [23] into self-supervised networks by our NAC for effective image denoising.

TABLE I

SUMMARY OF REPRESENTATIVE NETWORKS FOR IMAGE DENOISING. **S.**: SUPERVISED NETWORKS. **U.**: UNSUPERVISED NETWORKS. **SS.**: SELF-SUPERVISED NETWORKS. **PUB.**: PUBLICATION. **INT.**: INTERNAL IMAGE PRIORS. **EXT.**: EXTERNAL IMAGE PRIORS. **STAT.**: STATISTICS. THE NETWORKS WITH "✓" ARE ABLE TO LEARN THE NOISE STATISTICS FROM TRAINING DATA

Type	Method	Year'Pub.	Image Prior	Noise Stat.
S.	DnCNN [41]	17'TIP	Ext.	✓
	CBDNet [14]	19'CVPR	Ext.	✓
U.	Noise2Noise [22]	18'ICML	Ext.	✓
	GAN-CNN [7]	18'CVPR	Ext.	✓
	Noise2Void [19]	19'CVPR	Ext.	
SS.	Deep Image Prior [23]	18'CVPR	Int.	
	Noise2Self [3]	19'ICML	Ext.	
	Self-Supervised [20]	19'NeurIPS	Ext.	
	Noisy-As-Clean (Ours)	20'TIP	Int.	✓

- Experiments on synthetic and real-world benchmarks show that, on weak noise, the DnCNN and ResNet in [23] trained by our NAC achieve comparable or even better performance than the comparison denoising networks.

The remaining parts of this paper are organized as follows. In §II, we introduce the related work. In §III, we present the theoretical background of our NAC strategy for self-supervised image denoising. In §IV, we implement the DnCNN [41] and ResNet used in [23] as self-supervised networks by our NAC. Extensive experiments are conducted in §V demonstrate that, the DnCNN and ResNet networks trained by our NAC achieve comparable or even better performance than previous supervised image denoising networks on benchmark synthetic and real-world datasets. Conclusion is given in §VI.

II. RELATED WORK

In Table I, we summarize several state-of-the-art supervised [14], [41], unsupervised [7], [19], [22] and self-supervised [3], [20], [23] networks, image priors, and noise statistics. In this work, to bridge the *domain gap problem*, we propose a "Noisy-As-Clean" strategy to learn the image-specific internal prior and noise statistics directly from the corrupted test image.

A. Supervised Denoising Networks

are trained with plenty pairs of noisy and clean images. This category of networks can learn external image priors and noise statistics from the training data. Several methods [26], [30], [41] have been developed with achieving promising performance on AWGN noise removal, where the statistics of training and test noise are similar. However, due to the aforementioned *domain gap problem*, the performance of these networks degrade severely on real-world noisy images [28], [29], [35].

B. Unsupervised and Self-Supervised Denoising Networks

are developed to remove the need on plenty of clean images. Along this direction, Noise2Noise (N2N) [22] trains

the network between pairs of corrupted images with the same scene, but independently sampled noise. This work is feasible to learn external image priors and noise statistics from the training data. However, in real-world scenarios, it is difficult to collect large amounts of paired images with independent corruption for training. Noise2Void (N2V) [19] predicts a pixel from its surroundings by learning blind-spot networks, but it still suffers from the domain gap on image priors between the training images and test images. This work assumes that the corruption is zero-mean and independent between pixels. However, as mentioned in Noise2Self (N2S) [3], N2V [19] significantly degrades the training efficiency and denoising performance at test time. Recently, Deep Image Prior (DIP) [23] reveals that the network structure can resonate with the natural image priors, and can be utilized in image restoration without external images. However, it is not practical to select a suitable network and early-stop its training at right moments for each corrupted image. Self-supervised denoisers [3], [20] employ explicit corruption models, and train the networks only with the corrupted image itself. In this work, we utilize the helpful noise model to learn self-supervised denoising networks for real-world image denoising.

C. Internal and External Image Priors

are widely used for diverse image restoration tasks [36], [39], [40]. Internal priors are directly learned from the input test image itself, such as the multi-scale priors [11], [12], [34], image-specific details [24], [42], and non-local self similarity [16], [38], [39]. The external ones are learned on external natural images [37], [40], [43]. Internal priors are adaptive to its image contents, but somewhat affected by the corruptions [12], [42]. By contrast, the external priors are effective for restoring images with general contents, but may not be optimal for specific test image [8], [40], [43].

D. Noise Statistics

is of key importance for image denoising. The AWGN noise is one typical noise with widespread study. Recently, researchers shift more attention to the realistic noise produced in camera sensors [1], [29], which is usually modeled as mixed Poisson and Gaussian distribution [13]. The Poisson component mainly comes from the irregular photons hitting the sensor [25], while Gaussian noise is majorly produced by dark current [28]. Though performing well on the synthetic noise being trained with, supervised denoisers [14], [26], [41] still suffer from the *domain gap problem* when processing the real-world noisy images.

III. Theoretical Background of "Noisy-As-Clean"

Training a supervised network f_θ (parameterized by θ) requires many pairs $\{(\mathbf{y}_i, \mathbf{x}_i)\}$ of noisy image $\mathbf{y}_i$ and clean image $\mathbf{x}_i$, by minimizing an empirical loss function $\mathcal{L}$ as

$$\sum_{i=1} \mathcal{L}(f_\theta(\mathbf{y}_i), \mathbf{x}_i). \tag{1}$$

Assuming that the probability of occurrence for pair $(\mathbf{y}_i, \mathbf{x}_i)$ is $p(\mathbf{y}_i, \mathbf{x}_i)$, then statistically we have

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_{i=1} p(\mathbf{y}_i, \mathbf{x}_i)\mathcal{L}(f_\theta(\mathbf{y}_i), \mathbf{x}_i) \\ &= \arg\min_\theta \mathbb{E}_{(\mathbf{y},\mathbf{x})}[\mathcal{L}(f_\theta(\mathbf{y}), \mathbf{x})],\end{aligned} \tag{2}$$

where $\mathbf{y}$ and $\mathbf{x}$ are random variables of noisy and clean images, respectively. The paired variables $(\mathbf{y}, \mathbf{x})$ are dependent, and their relationship is $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$, where $\mathbf{n}_o$ is the random variable of *observed* noise. By exploring the dependence of $p(\mathbf{y}_i, \mathbf{x}_i) = p(\mathbf{x}_i)p(\mathbf{y}_i|\mathbf{x}_i)$, Eqn. (2) is equivalent to

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_{i=1} p(\mathbf{x}_i)p(\mathbf{y}_i|\mathbf{x}_i)\mathcal{L}(f_\theta(\mathbf{y}_i), \mathbf{x}_i) \\ &= \arg\min_\theta \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathcal{L}(f_\theta(\mathbf{y}), \mathbf{x})]].\end{aligned} \tag{3}$$

This indicates that the network f_θ can minimize the loss function by solving Eqn. (3) separately for each clean image.

Different with the "zero-mean" assumption in [19], [22], here we study a more practical assumption on noise statistics, i.e., *the expectation* $\mathbb{E}[\mathbf{x}]$ *and variance* $\mathrm{Var}[\mathbf{x}]$ *of signal intensity are much stronger than those of noise* $\mathbb{E}[\mathbf{n}_o]$ *and* $\mathrm{Var}[\mathbf{n}_o]$ (negligible but not necessarily zero):

$$\mathbb{E}[\mathbf{x}] \gg \mathbb{E}[\mathbf{n}_o], \ \ \mathrm{Var}[\mathbf{x}] \gg \mathrm{Var}[\mathbf{n}_o]. \tag{4}$$

This is actually valid in real-world scenarios, since we can clearly observe the contents in most real photographs, *with little influence of the noise*. The noise therein is often modeled by zero-mean Gaussian or mixed Poisson and Gaussian (for realistic noise). Hence, the noisy image $\mathbf{y}$ should have similar expectation with the clean image $\mathbf{x}$:

$$\mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{x} + \mathbf{n}_o] = \mathbb{E}[\mathbf{x}] + \mathbb{E}[\mathbf{n}_o] \approx \mathbb{E}[\mathbf{x}]. \tag{5}$$

Now we add *simulated* noise $\mathbf{n}_s$ to the *observed* noisy image $\mathbf{y}$, and generate a new noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$. We assume that $\mathbf{n}_s$ is statisticly close to $\mathbf{n}_o$, i.e., $\mathbb{E}[\mathbf{n}_s] \approx \mathbb{E}[\mathbf{n}_o]$ and $\mathrm{Var}[\mathbf{n}_s] \approx \mathrm{Var}[\mathbf{n}_o]$. Then we have

$$\mathbb{E}[\mathbf{z}] \gg \mathbb{E}[\mathbf{n}_s], \ \ \mathrm{Var}[\mathbf{z}] \gg \mathrm{Var}[\mathbf{n}_s]. \tag{6}$$

Therefore, the *simulated* noisy image $\mathbf{z}$ has similar expectation with the *observed* noisy image $\mathbf{y}$:

$$\mathbb{E}[\mathbf{z}] = \mathbb{E}[\mathbf{y} + \mathbf{n}_s] \approx \mathbb{E}[\mathbf{y}]. \tag{7}$$

By the *Law of Total Expectation* [4], we have

$$\mathbb{E}_{\mathbf{y}}[\mathbb{E}_{\mathbf{z}}[\mathbf{z}|\mathbf{y}]] = \mathbb{E}[\mathbf{z}] \approx \mathbb{E}[\mathbf{y}] = \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{\mathbf{y}}[\mathbf{y}|\mathbf{x}]]. \tag{8}$$

Since the loss function $\mathcal{L}$ (usually ℓ_2) and the conditional probability density functions $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{y})$ are all *continuous everywhere*, the optimal network parameters θ^* of Eqn. (3) changes little with the addition of negligible noise $\mathbf{n}_o$ or $\mathbf{n}_s$. With Eqns. (4)-(8), when the $\mathbf{x}$-conditioned expectation of $\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathcal{L}(f_\theta(\mathbf{y}), \mathbf{x})]$ are replaced with the $\mathbf{y}$-conditioned expectation of $\mathbb{E}_{\mathbf{z}|\mathbf{y}}[\mathcal{L}(f_\theta(\mathbf{z}), \mathbf{y})]$, f_θ obtains similar $\mathbf{y}$-conditioned optimal parameters θ^*:

$$\begin{aligned}&\arg\min_\theta \mathbb{E}_{\mathbf{y}}[\mathbb{E}_{\mathbf{z}|\mathbf{y}}[\mathcal{L}(f_\theta(\mathbf{z}), \mathbf{y})]] \\ &\qquad \approx \arg\min_\theta \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathcal{L}(f_\theta(\mathbf{y}), \mathbf{x})]] = \theta^*.\end{aligned} \tag{9}$$

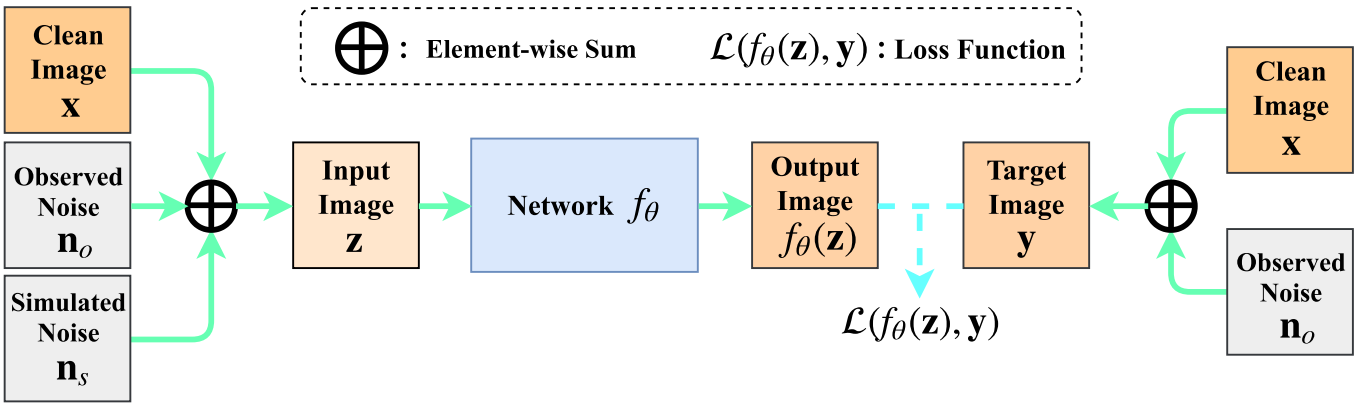


Fig. 2. **Proposed "Noisy-As-Clean" strategy for training self-supervised image denoising networks**. In our NAC strategy, we take the *observed* noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ as the "clean" target, and take the *simulated* noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$ as the input. We do not regard the clean image $\mathbf{x}$ as target. After training, the inference is performed on the target noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$.

The network f_θ minimizes the loss function $\mathcal{L}$ for each input image pair separately, which equals to minimize it on all finite pairs of images. Through simple manipulations, Eqn. (9) is equivalent to

$$\arg\min_\theta \sum_{i=1} p(\mathbf{y}_i)p(\mathbf{z}_i|\mathbf{y}_i)\mathcal{L}(f_\theta(\mathbf{z}_i), \mathbf{y}_i) = \arg\min_\theta \mathbb{E}_{\mathbf{y}}[\mathbb{E}_{\mathbf{z}|\mathbf{y}}[\mathcal{L}(f_\theta(\mathbf{z}), \mathbf{y})]] \approx \theta^*. \quad (10)$$

By exploring the dependence of $p(\mathbf{z}_i, \mathbf{y}_i) = p(\mathbf{y}_i)p(\mathbf{z}_i|\mathbf{y}_i)$, Eqn. (10) is equivalent to

$$\arg\min_\theta \mathbb{E}_{(\mathbf{z},\mathbf{y})}[\mathcal{L}(f_\theta(\mathbf{z}), \mathbf{y})] = \arg\min_\theta \sum_{i=1} p(\mathbf{z}_i, \mathbf{y}_i)\mathcal{L}(f_\theta(\mathbf{z}_i), \mathbf{y}_i) \approx \theta^*. \quad (11)$$

A Simple But Useful Observation Is: *as long as the noise is weak, the optimal parameters of self-supervised network trained on noisy image pairs* $\{(\mathbf{z}_i, \mathbf{y}_i)\}$ *are very close to the optimal parameters of the supervised networks trained on pairs of noisy and clean images* $\{(\mathbf{y}_i, \mathbf{x}_i)\}$. In Figure 3, we explain our NAC strategy and illustrate this observation through an example on the image "Test004" from the BSD68 dataset: The clean image $\mathbf{x}$ in (a) is firstly corrupted by observed AWGN noise $\mathbf{n}_o$ with $\sigma = 5$. Then we add simulated AWGN noise $\mathbf{n}_s$ also with $\sigma = 5$ to the corrupted image $\mathbf{x}+\mathbf{n}_o$ in (b), and obtain a doubly corrupted image $\mathbf{x} + \mathbf{n}_o + \mathbf{n}_s$ in (c). The DnCNN [41] with our NAC strategy, named as DnCNN+NAC, is trained with the doubly corrupted image $\mathbf{x}+\mathbf{n}_o+\mathbf{n}_s$ in (c) as input and the corrupted image $\mathbf{x}+\mathbf{n}_o$ in (b) as target. The final training output is plotted in (d), with similar PSNR and SSIM [33] results as the corrupted image $\mathbf{x} + \mathbf{n}_o$ in (b). Then the DnCNN+NAC network learned on (b) and (c) is directly employed to perform inference on the corrupted image $\mathbf{x} + \mathbf{n}_o$ in (b), and produces the testing output in (e). When compared to DnCNN [41], our DnCNN+NAC achieves much higher PSNR and SSIM results on the corrupted image (b). The estimated simulated noise $\mathbf{n}_s$ and observed noise $\mathbf{n}_o$ in training and test stages are plotted in (g) and (h), respectively. One can see that they are visually in similar noise statistics.

Consistency of Noise Statistics: Since our contexts are the real-world scenarios, the noise can be modeled by mixed Poisson and Gaussian distribution [13]. Fortunately, both the two distributions are linear additive, i.e., the addition variable of two Poisson (or Gaussian) distributed variables are still Poisson (or Gaussian) distributed. Assume that the observed (simulated) noise $\mathbf{n}_o$ ($\mathbf{n}_s$) follows a mixed $\mathbf{x}$-dependent ($\mathbf{y}$-dependent) Poisson distribution parameterized by λ_o (λ_s) and Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma_o^2)$ ($\mathcal{N}(\mathbf{0}, \sigma_s^2)$), i.e.,

$$\begin{aligned} \mathbf{n}_o &\sim \mathbf{x} \odot \mathcal{P}(\lambda_o) + \mathcal{N}(\mathbf{0}, \sigma_o^2), \\ \mathbf{n}_s &\sim \mathbf{y} \odot \mathcal{P}(\lambda_s) + \mathcal{N}(\mathbf{0}, \sigma_s^2) \\ &\approx \mathbf{x} \odot \mathcal{P}(\lambda_s) + \mathcal{N}(\mathbf{0}, \sigma_s^2), \end{aligned} \quad (12)$$

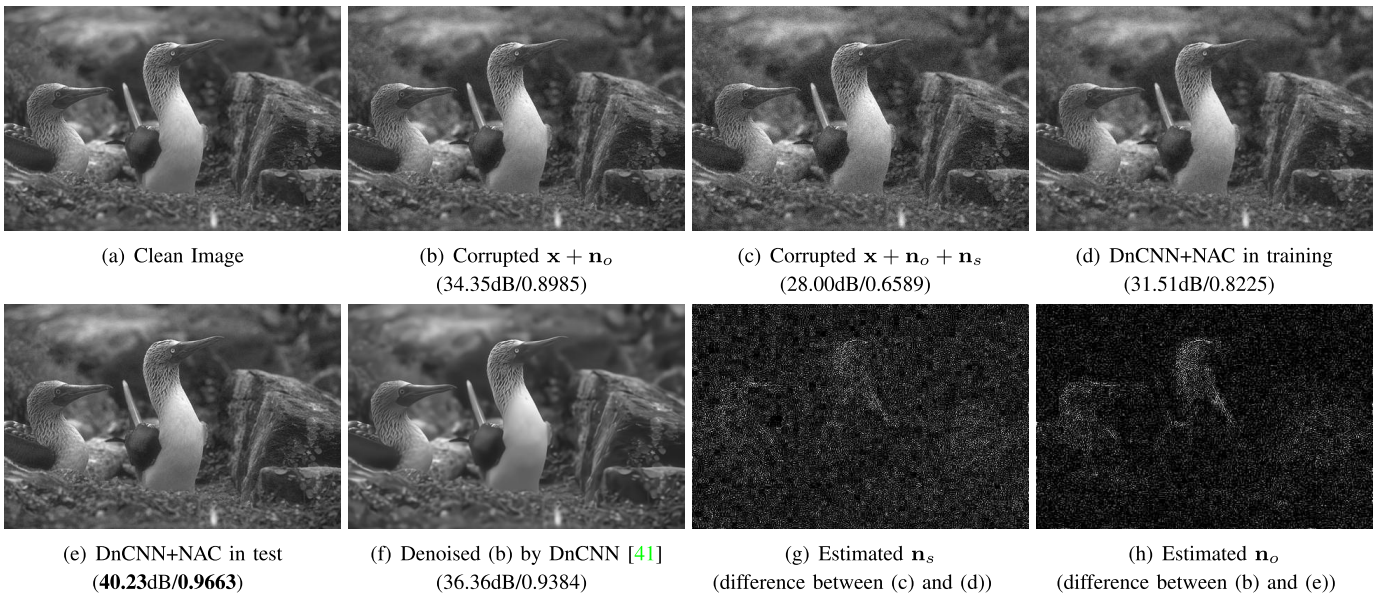
where $\mathbf{x} \odot \mathcal{P}(\lambda_o)$ and $\mathbf{y} \odot P(\lambda_s)$ indicates that the noise $\mathbf{n}_o$ and $\mathbf{n}_s$ are element-wisely dependent on $\mathbf{x}$ and $\mathbf{y}$, respectively. The "$\approx$" is valid if we assume that the observed noise $\mathbf{n}_o$ is "weak" when compared to the signal $\mathbf{x}$. To this end, we have

$$\mathbf{n}_o+\mathbf{n}_s \sim \mathbf{x} \odot \mathcal{P}(\lambda_o+\lambda_s)+\mathcal{N}(\mathbf{0}, \sigma_o^2+\sigma_s^2+2\rho\sigma_o\sigma_s), \quad (13)$$

where ρ is the correlation between $\mathbf{n}_o$ and $\mathbf{n}_s$ ($\rho = 0$ if they are independent). This indicates that the summed noise variable $\mathbf{n}_o+\mathbf{n}_s$ still follows a mixed $\mathbf{x}$ dependent Poisson and Gaussian distribution, guaranteeing the consistency in noise statistics between the *observed* realistic noise and the *simulated* noise. As will be validated by the experiments (§V), this property makes our NAC strategy consistently effective on different noise removal tasks.

IV. Learning Self-Supervised Denoising Networks by "Noisy-As-Clean"

Here, we propose to learn self-supervised denoising networks with our "Noisy-As-Clean" (NAC) strategy. We employ the DnCNN [41] and ResNet in DIP [23] as our baseline, and call the self-supervised networks as DnCNN+NAC and ResNet+NAC, respectively. Note that we only need the *observed* noisy image $\mathbf{y}$ to generate noisy image pairs $\{(\mathbf{z}, \mathbf{y})\}$ with *simulated* noise $\mathbf{n}_s$, as illustrated in Figure 2.



(a) Clean Image | (b) Corrupted $\mathbf{x}+\mathbf{n}_o$ (34.35dB/0.8985) | (c) Corrupted $\mathbf{x}+\mathbf{n}_o+\mathbf{n}_s$ (28.00dB/0.6589) | (d) DnCNN+NAC in training (31.51dB/0.8225)

(e) DnCNN+NAC in test (**40.23**dB/**0.9663**) | (f) Denoised (b) by DnCNN [41] (36.36dB/0.9384) | (g) Estimated $\mathbf{n}_s$ (difference between (c) and (d)) | (h) Estimated $\mathbf{n}_o$ (difference between (b) and (e))

Fig. 3. **An example to illustrate the pipeline of our NAC strategy based image denoising**. The image is "*Test004*" from the *BSD68* dataset. The observed noise $\mathbf{n}_o$ and simulated noise $\mathbf{n}_s$ are additive white Gaussian noise with $\sigma = 5$. (a) The clean image $\mathbf{x}$. (b) The corrupted image $\mathbf{x}+\mathbf{n}_o$ (training target of our DnCNN+NAC). (c) The doubly corrutped image $\mathbf{x}+\mathbf{n}_o+\mathbf{n}_s$. (d) The output of training DnCNN in our NAC strategy, with input is the doubly corrupted image (c) and target is the corrupted image (b). (e) The output of our image-specific DnCNN+NAC tested on (b). (f) The output of DnCNN tested on (b). (g) The estimated $\mathbf{n}_s$ is the difference between (c) and (d). (h) The estimated $\mathbf{n}_o$ is the difference between (b) and (e). Note that the values in images (g) and (h) are amplified by 10 times for better visualization. PSNR and SSIM results of corresponding images are provided for objective references.

A. Training Self-Supervised Networks by Our NAC

For real-world images captured by camera sensors, one can hardly distinguish the realistic noise from the signal. The signal intensity $\mathbf{x}$ is usually stronger than the noise intensity. That is, the expectation of the observed realistic noise $\mathbf{n}_o$ is usually much smaller than that of the latent clean image $\mathbf{x}$. If we train an image-specific network for the new noisy image $\mathbf{z}$ and regard the original noisy image $\mathbf{y}$ as the ground-truth image, then the trained image-specific network basically joint learn the image-specific prior and noise statistics. It has the capacity to remove the noise $\mathbf{n}_s$ from the new noisy image $\mathbf{z}$. Then if we perform denoising on the original noisy image $\mathbf{y}$, the observed noise $\mathbf{n}_o$ can be well-removed. Note that we *do not* use the clean image $\mathbf{x}$ as "ground-truth" in training the DnCNN+NAC and ResNet+NAC networks.

B. Training Blind Denoising Networks

Most of existing supervised denoising networks train a specific model to process a fixed noise pattern [5], [26], [28]. To tackle unknown noise, one feasible solution for these networks is to assume the noise as AWGN and estimate its noise deviation. The corresponding noise is removed by using the networks trained with the estimated level. But this strategy largely degrades the denoising performance when the noise deviation is not estimated accurately. Besides, this solution can hardly deal with realistic noise, which is usually not AWGN, captured on real photographs. In order to be effective on removing realistic noise, the self-supervised networks by our NAC are feasible to blindly remove the unknown noise from real photographs. Inspired by [14], [41], we propose to train a blind version of DnCNN+NAC and ResNet+NAC networks by using the AWGN noise within a range of levels (e.g., $[0, 55]$) for removing unknown AWGN noise. We also train blind ResNet+NAC with mixed AWGN and Poisson noise (both within a range of intensities) for removing the realistic noise. More details will be explained in §V-B.

C. Testing

is performed by directly regarding an *observed* noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ as input. We only test the image $\mathbf{y}$ once. The denoised image can be represented as $\hat{\mathbf{y}} = f_{\theta^*}(\mathbf{y})$, with which the objective metrics, e.g., PSNR and SSIM [33], can be computed with the clean image $\mathbf{x}$.

D. Implementation Details

We employ the DnCNN [41] or the ResNet (used in DIP [23]) as the backbones, and turn them into self-supervised networks by our NAC strategy, which are named as DnCNN+NAC or ResNet+NAC, respectively. The DnCNN contains 17 layers of convolution, Batch Normalization (BN) [17], and Rectified Linear Units (ReLU) activation operator [27]. To accommodate DnCNN with our NAC strategy, we set the output of DnCNN+NAC as the denoised image, not the residual noise in DnCNN [41]. We observe no difference between the results on PSNR, SSIM [33], and visual quality by employing these two types of outputs in our experiments. As DnCNN, the parameters of DnCNN+NAC are initialized from a pretrained ResNet. As used in [23], the ResNet in our ResNet+NAC includes 10 residual blocks, each containing two convolutional layers followed by a BN [17] and a ReLU [27] after the first BN. The parameters are randomly initialized without being pretrained. For both baselines,

TABLE II

AVERAGE PSNR (dB) AND SSIM [33] RESULTS OF DIFFERENT METHODS ON SET12 DATASET CORRUPTED BY AWGN NOISE. THE FIRST, SECOND, AND THIRD BEST RESULTS ARE HIGHLIGHTED IN **RED**, **BLUE**, AND **BOLD**, RESPECTIVELY

Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$		$\sigma = 25$	
Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
BM3D [10]	38.07	0.9580	34.40	0.9234	32.38	0.8957	31.00	0.8717	29.97	0.8503
DnCNN [41]	38.76	0.9633	34.78	0.9270	32.86	0.9027	31.45	0.8799	30.43	0.8617
N2N [22]	**39.72**	0.9665	36.18	0.9446	33.99	0.9149	**32.10**	0.8788	30.72	0.8446
DIP [23]	32.49	0.9344	31.49	0.9299	29.59	0.8636	27.67	0.8531	25.82	0.7723
N2V [19]	27.06	0.8174	26.79	0.7859	26.12	0.7468	25.89	0.7405	25.01	0.6564
DnCNN+NAC	**43.17**	**0.9817**	**37.16**	**0.9336**	33.64	0.8697	31.15	0.8024	29.22	0.7382
Blind DnCNN+NAC	**43.16**	**0.9817**	**37.14**	**0.9333**	33.63	0.8693	31.14	0.8018	29.21	0.7376
ResNet+NAC	**39.99**	**0.9820**	**36.55**	**0.9569**	**34.24**	**0.9277**	**32.46**	**0.8961**	**31.08**	**0.8654**
Blind ResNet+NAC	38.48	**0.9805**	**36.65**	**0.9564**	**34.77**	**0.9275**	**33.13**	**0.9024**	**31.78**	**0.8802**

TABLE III

AVERAGE PSNR (dB) AND SSIM [33] RESULTS OF DIFFERENT METHODS ON *BSD68* DATASET CORRUPTED BY AWGN NOISE. THE FIRST, SECOND, AND THIRD BEST RESULTS ARE HIGHLIGHTED IN **RED**, **BLUE**, AND **BOLD**, RESPECTIVELY

Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$		$\sigma = 25$	
Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
BM3D [10]	37.59	0.9640	33.32	0.9163	31.07	0.8720	29.62	0.8342	28.57	0.8017
DnCNN [41]	38.07	**0.9695**	33.88	**0.9270**	31.73	0.8706	**30.27**	**0.8563**	**29.23**	**0.8278**
N2N [22]	**38.58**	0.9627	34.07	0.9200	**31.81**	**0.8770**	30.14	0.8550	28.67	0.8123
DIP [23]	29.74	0.8435	28.16	0.8310	27.07	0.7867	25.80	0.7205	24.63	0.6680
N2V [19]	26.70	0.7915	26.39	0.7621	25.77	0.7126	25.41	0.6678	24.83	0.6305
DnCNN+NAC	**40.21**	**0.9674**	**34.21**	0.8913	30.72	0.8044	28.25	0.7230	26.34	0.6515
Blind DnCNN+NAC	**40.20**	**0.9674**	**34.21**	0.8911	30.71	0.8041	28.24	0.7227	26.33	0.6511
ResNet+NAC	**39.00**	**0.9707**	**34.60**	**0.9324**	**32.13**	**0.8942**	**30.47**	**0.8636**	**28.96**	**0.8185**
Blind ResNet+NAC	38.26	0.9605	**34.26**	**0.9266**	**32.06**	**0.8919**	**30.50**	**0.8609**	**29.33**	**0.8327**

the optimizer is Adam [18] with default parameters. The learning rate is fixed at 0.001 in all experiments. We use the ℓ_2 loss function. For each test image, we only train the DnCNN+NAC in 100 epochs, while the original DnCNN is trained with 180 epochs. The ResNet+NAC is trained in 1000 epochs for each test image, the same as that in DIP [23]. As suggested by DnCNN [41] and DIP [23], we employ 4 rotations {0°, 90°, 180°, 270°} combined with 2 mirror (vertical and horizontal) reflections, resulting in totally 8 transformations for data augmentation. We implement the DnCNN+NAC and ResNet+NAC networks in PyTorch.

V. EXPERIMENTS

In this section, we evaluate the performance of our "Noisy-As-Clean" (NAC) networks on image denoising. In all experiments, we train a denoising network using only the noisy test image $\mathbf{y}$ as the target, and using the *simulated* noisy image $\mathbf{z}$ as the input. For all comparison methods, the source codes or trained models are downloaded from the corresponding authors' websites. We use the default parameter settings, unless otherwise specified. The PSNR, SSIM [33], and visual quality of different methods are used to evaluate the comparison. We first test with synthetic noise such as additive white Gaussian noise (AWGN) in §V-A, continue to perform blind image denoising in §V-B, and finally tackle the realistic noise in §V-C. In §V-D, we conduct comprehensive ablation studies to gain deeper insights into our NAC strategy.

A. Synthetic Noise Removal With Known Noise

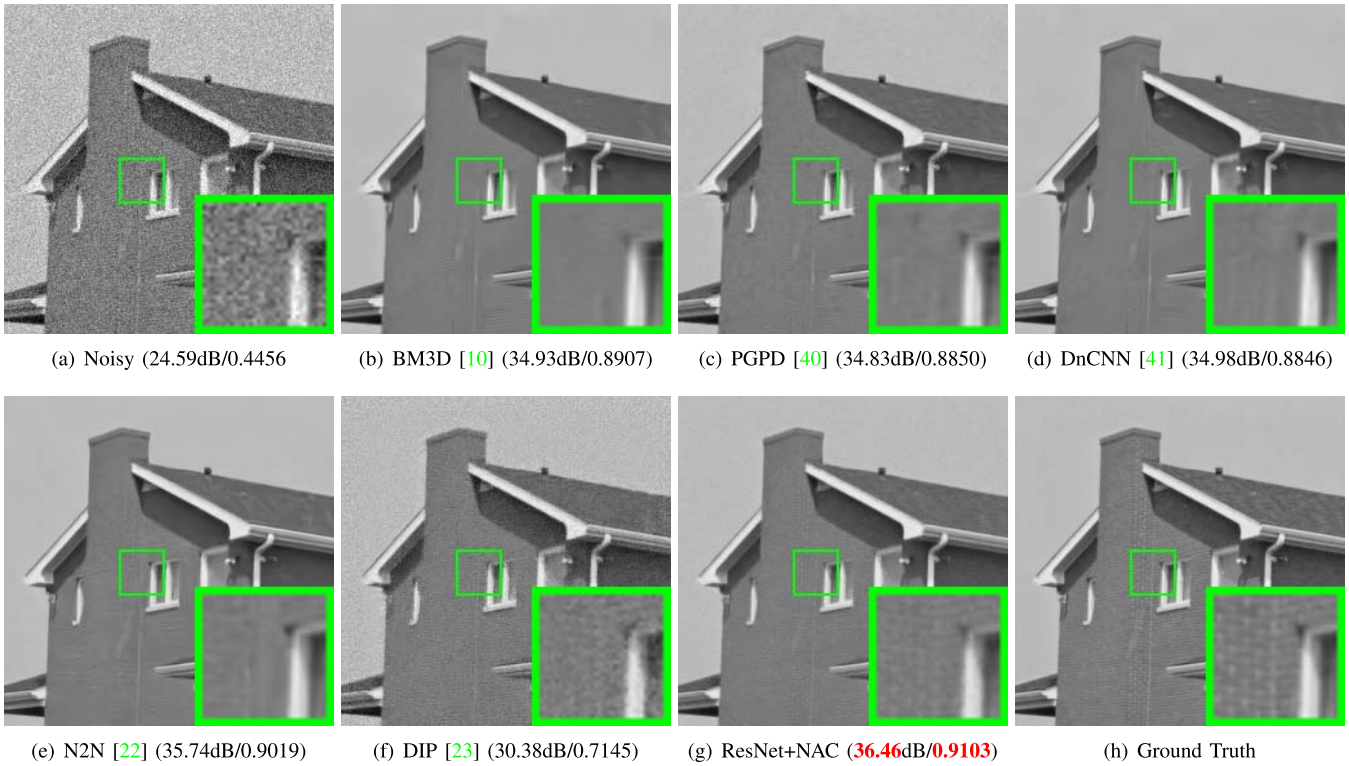
We evaluate the DnCNN+NAC and ResNet+NAC networks on images corrupted by synthetic AWGN noise. More results on signal dependent Poisson noise and mixed Poisson-AWGN noise are provided in the *Supplementary File*.

1) Training Self-Supervised Networks: Here, we train an image-specific denoising network using the *observed* noisy test image $\mathbf{y}$ as the target, and the *simulated* noisy image $\mathbf{z}$ as the input. Each *observed* noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ is generated by adding the *observed* noise $\mathbf{n}_o$ to the clean image $\mathbf{x}$. The *simulated* noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$ is generated by adding *simulated* noise $\mathbf{n}_s$ to *observed* noisy image $\mathbf{y}$.

2) Comparison Methods: We compare DnCNN+NAC and ResNet+NAC networks with state-of-the-art image denoising methods [10], [22], [41]. On AWGN noise, we compare with BM3D [10], DnCNN [41], Noise2Noise (N2N) [22], Deep Image Prior (DIP) [23], and Noise2Void (N2V) [19].

3) Test Datasets: We evaluate the comparison methods on the *Set12* and *BSD68* datasets, which are widely tested by supervised denoising networks [26], [41] and previous methods [10], [40]. The *Set12* dataset contains 12 images of sizes 512×512 or 256×256, while the *BSD68* dataset contains 68 images of different sizes.

4) Results on AWGN Noise: with noise levels (standard deviation, or std) of $\sigma \in \{5, 10, 15, 20, 25\}$ are provided here. The *observed* noise $\mathbf{n}_o$ is AWGN with std of σ, while the *simulated* noise $\mathbf{n}_s$ is with the same σ as that of $\mathbf{n}_o$. The comparison results are listed in Tables II and III. It can be



(a) Noisy (24.59dB/0.4456) (b) BM3D [10] (34.93dB/0.8907) (c) PGPD [40] (34.83dB/0.8850) (d) DnCNN [41] (34.98dB/0.8846)

(e) N2N [22] (35.74dB/0.9019) (f) DIP [23] (30.38dB/0.7145) (g) ResNet+NAC (**36.46**dB/**0.9103**) (h) Ground Truth

Fig. 4. **Denoised images and PSNR/SSIM results of "*House*" in *Set12* by different methods**. The images are corrupted by AWGN noise with $\sigma = 15$. The best results on PSNR and SSIM are highlighted in **bold**.

seen that, DnCNN+NAC achieves better PSNR and SSIM results than those of the original DnCNN when $\sigma = 5, 10$. Note that DnCNN are supervised networks trained offline on the *BSD400* dataset, while the variant DnCNN+NAC network is trained online for each corrupted image. Besides, the blind version of DnCNN+NAC achieves negligible performance drop when compared to the DnCNN+NAC, which is consistent with [41]. On the other side, the ResNet+NAC networks achieve comparable or better performance on PSNR and SSIM [33] than BM3D [10] and DnCNN [41], especially when the noise levels are weak ($\sigma = 5, 10$). Besides, our ResNet+NAC networks outperform the other unsupervised and self-supervised networks such as N2N [22], DIP [23], and N2V [19] by a large margin on PSNR and SSIM [33]. In Figures 4 and 5, we provide the visual comparisons of the denoised images by the competing methods. One can see that the ResNet+NAC networks produce better image quality and higher PSNR/SSIM results than the comparison methods.

B. Synthetic Noise Removal With Unknown Noise

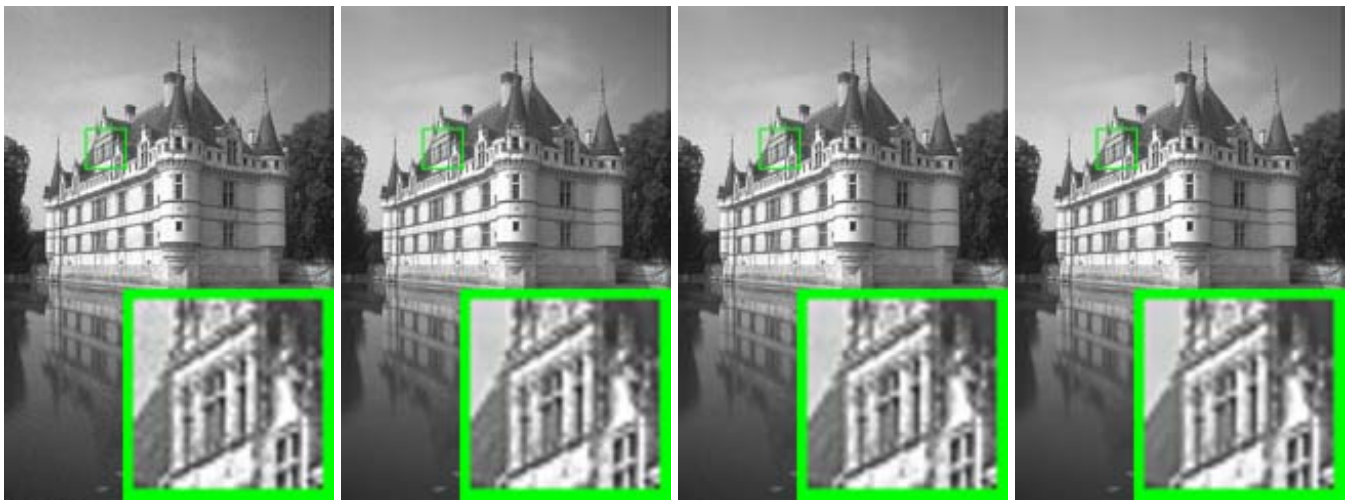
To deal with unknown noise, we propose to train blind versions of the DnCNN [41] and ResNet in [23] by our NAC strategy. Here, we test the Blind DnCNN+NAC and Blind ResNet+NAC networks on AWGN noise with unknown noise deviation. We use the same training strategy, comparison methods, and test datasets as in §V-A.

1) Training Blind Networks: We train the Blind DnCNN+NAC and Blind ResNet+NAC networks on the corrupted test image degraded *again* by AWGN noise with unknown noise levels (deviations). The noise levels are randomly sampled in Gaussian distribution within $[0, 55]$. We also test on noise levels in uniform distribution and obtain similar results. We repeat the training of DnCNN+NAC and ResNet+NAC networks on the test image with different deviations.

2) Results on Blind Denoising: For the same test image, we add to it the AWGN noise whose deviation is also in $\{5, 10, 15, 20, 25\}$. The blindly trained DnCNN+NAC and ResNet+NAC networks are directly utilized to denoise the test image without estimating its deviation. The results are also listed in Tables II and III. We observe that, the Blind ResNet+NAC networks trained on AWGN noise with unknown levels can achieve even better PSNR and SSIM [33] results than the ResNet+NAC networks trained on specific noise levels. Note that on *BSD68*, the ResNet+NAC networks achieve higher PSNR and SSIM results than DnCNN [41]. This demonstrates the effectiveness of our ResNet+NAC networks on blind image denoising. With the success on blind image, next we will turn to real-world image denoising, in which the noise is also unknown and very complex.

C. Practice on Real Photographs

With the promising performance on blind image denoising, here we tackle the realistic noise for practical applications. The *observed* realistic noise $\mathbf{n}_o$ can be roughly modeled as mixed Poisson noise and AWGN noise [13], [14]. Hence, for each *observed* noisy image $\mathbf{y}$, we generate the *simulated* noise $\mathbf{n}_s$



(a) Noisy (34.15dB/0.8416) (b) BM3D [10] (38.20dB/0.9569) (c) PGPD [40] (38.02dB/0.9524) (d) DnCNN [41] (38.64dB/0.9559)

(e) N2N [22] (39.63dB/0.9682) (f) DIP [23] (27.22dB/0.8794) (g) ResNet+NAC (**39.89**dB/**0.9693**) (h) Ground Truth

Fig. 5. **Denoised images and PSNR/SSIM results of "*Test003* " in *BSD68* by different methods**. The images are corrupted by AWGN noise with $\sigma = 5$. The best results on PSNR and SSIM are highlighted in **bold**.

by sampling the **y**-dependent Poisson part and the independent AWGN noise.

1) Training Blind ResNet+NAC Networks: is also performed for each test image, i.e., the *observed* noisy image **y**. In real-world scenarios, each *observed* noisy image **y** is corrupted without knowing the specific noise statistics of the *observed* noise $\mathbf{n}_o$. Therefore, the *simulated* noise $\mathbf{n}_s$ is directly estimated on **y** as mixed **y**-dependent Poisson and AWGN noise. For each transformation image in data augmentation, the Poisson noise is randomly sampled with the parameter λ in $0 < \lambda \leq 25$, and the AWGN noise is randomly sampled with the noise level σ in $0 < \sigma \leq 25$.

2) Comparison Methods: We compare with state-of-the-art methods on real-world image denoising, including CBM3D [9], the commercial software Neat Image [2], two supervised networks DnCNN+ [41] and CBDNet [14], and two unsupervised networks GCBD [7] and Noise2Noise [22], and the self-supervised network DIP [23]. Note that DnCNN+ [41] and CBDNet [14] are two state-of-the-art supervised networks for real-world image denoising, and DnCNN+ is an improved extension of DnCNN [41] with better performance (the authors of DnCNN+ provide us the models/results of DnCNN+).

3) Test Datasets: We evaluate the comparison methods on the *Cross-Channel* (*CC*) dataset [28] and *DND* dataset [29]. The *CC* dataset [28] includes noisy images of 11 static scenes captured by Canon 5D Mark 3, Nikon D600, and Nikon D800 cameras. The noisy images are collected under a highly controlled indoor environment. Each scene is shot 500 times using the same settings. The average of the 500 shots is taken as "ground-truth". We use the default 15 images of size 512×512 cropped by the authors to evaluate different image denoising methods. The *DND* dataset [29] contains 50 scenarios captured by Sony A7R, Olympus E-M10, Sony RX100 IV, and Huawei Nexus 6P. Each scene is cropped to 20 bounding boxes of 512×512 pixels, generating totally 1000 test images. The noisy images are collected under higher ISO values with shorter exposure times, while the "ground

TABLE IV

AVERAGE PSNR (dB) AND SSIM [33] OF DIFFERENT METHODS ON THE *CC* DATASET [28] AND THE *DND* DATASET [29]. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**. "NA" MEANS "NOT AVAILABLE" DUE TO UNAVAILABLE CODE (GCBD ON *CC* [28]) OR DIFFICULT EXPERIMENTS (DIP ON *DND* [29]). THE FIRST, SECOND, AND THIRD BEST RESULTS ARE HIGHLIGHTED IN **RED**, **BLUE**, AND **BOLD**, RESPECTIVELY

Dataset	Type	Traditional Methods		Supervised Networks		Unsupervised Networks		Self-supervised Networks	
	Method	**CBM3D** [9]	**NI** [2]	**DnCNN+** [41]	**CBDNet** [14]	**GCBD** [7]	**N2N** [22]	**DIP** [23]	**Blind ResNet+NAC**
CC [28]	PSNR↑	35.19	35.33	35.40	**36.44**	NA	35.32	**35.69**	**36.59**
	SSIM↑	0.9063	0.9212	0.9115	**0.9460**	NA	0.9160	**0.9259**	**0.9502**
DND [29]	PSNR↑	34.51	35.11	**37.90**	**38.06**	35.58	33.10	NA	**36.20**
	SSIM↑	0.8507	0.8778	**0.9430**	**0.9421**	0.9217	0.8110	NA	**0.9252**

truth" images are captured under lower ISO values with adjusted longer exposure times. The "ground truth" images are not released, but we can obtain the PSNR and SSIM results by submitting the denoised images to the https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb*DND*'s Website.

4) Comparison Results on PSNR and SSIM: are listed in Table IV. As can be seen, the ResNet+NAC networks achieve better performance than all previous denoising methods, including the CBM3D [9], the supervised networks DnCNN+ [41] and CBDNet [14], and the unsupervised networks GCBD [7], N2N [22], and DIP [23]. This demonstrates that the ResNet+NAC networks can indeed handle the complex, unknown, and realistic noise, and achieve better performance than supervised networks such as DnCNN+ [41] and CBDNet [14].

5) Qualitative Results: In Figures 6 and 7, we show the denoised images of our ResNet+NAC and the comparison methods on the images of "*5dmark3-iso3200-1*" from the *CC* dataset [28] and "*0017_3*" from the *DND* dataset [29], respectively. We observe that our self-supervised Blind ResNet+NAC is very effective on removing realistic noise from the real photograph. Besides, the Blind ResNet+NAC networks achieve competitive PSNR and SSIM results when compared with the other methods, including the supervised DnCNN+ [41] and CBDNet [14].

6) Speed: The work most similar to ours is Deep Image Prior (DIP) [23], which also trains an image-specific network for each test image. Averagely, DIP needs 603.9 seconds to process a 512×512 color image, on which our ResNet+NAC network needs 583.2 seconds (on an NVIDIA Titan X GPU).

D. Ablation Study

To further study our NAC strategy, we conduct more examination of our ResNet+NAC networks on image denoising. Specifically, we assess 1) differences of the ResNet+NAC from the ResNet in DIP [23]; 2) how the number of residual blocks and epochs influence the ResNet+NAC; 3) comparison with the "Oracle" performance of the ResNet+NAC networks; 4) performance of the ResNet+NAC on "strong" noise.

1) Differences From DIP [23]: Though the basic network in our work is the ResNet used in DIP [23], our ResNet+NAC network is essentially different from DIP on at least two aspects. First, our ResNet+NAC is a novel strategy for self-supervised learning of *adaptive network parameters* for the degraded image, while DIP aims to investigate *adaptive network structure* without learning the parameters.

TABLE V

AVERAGE PSNR (dB)/SSIM OF RESNET+NAC WITH DIFFERENT NUMBER OF BLOCKS ON *Set12* CORRUPTED BY AWGN NOISE ($\sigma = 15$)

# of Blocks	1	2	5	10	15
PSNR↑	33.58	33.85	34.14	34.24	34.26
SSIM↑	0.9161	0.9226	0.9272	0.9277	0.9272

Second, our ResNet+NAC learns a mapping from the synthetic noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$ to the noisy image $\mathbf{y}$, which approximates the mapping from the noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ to the clean image $\mathbf{x}$. But DIP maps a random noise map to the noisy image $\mathbf{y}$, and the denoised image is obtained during the process. Due to the two reasons, DIP needs early stop for different images, while our ResNet+NAC achieves more robust (and better) denoising performance than DIP on diverse images. In Figure 8, we plot the curves of training loss and test PSNR of DIP (a) and ResNet+NAC (b) networks in 10,000 epochs, on two images of "Cameraman" and "House". We observe that DIP needs early stop to select the best results, while our ResNet+NAC can stably achieve better denoising results within 1000 epochs.

2) Influence on the Number of Residual Blocks and Epochs: Our backbone network is the ResNet [23] with 10 residual blocks trained in 1000 epochs. Now we study how the number of residual blocks and epochs influence the performance of ResNet+NAC on image denoising. The experiments are performed on the *Set12* dataset corrupted by AWGN noise ($\sigma = 15$). From Table V, we observe that, with more residual blocks, the ResNet+NAC networks can achieve better PSNR and SSIM [33] results. And 10 residual blocks are enough to achieve satisfactory results. With more (e.g., 15) blocks, there is little improvement on PSRN and SSIM. Hence, we use 10 residual blocks the same as [23]. Then we study how the number of epochs influence the performance of ResNet+NAC on image denoising. From Table VI, one can see that on the *Set12* dataset corrupted by AWGN noise ($\sigma = 15$), with more training epochs, our ResNet+NAC networks achieve better PSNR and SSIM results, but with longer processing time.

3) Comparison With Oracle: We also study the "Oracle" performance of the ResNet+NAC networks. Roughly speaking, "Oracle" performance means the best performance a model can achieve (in our case, on image denoising trained only with the test image). In "Oracle", we train the ResNet+NAC networks on the pair of *observed* noisy

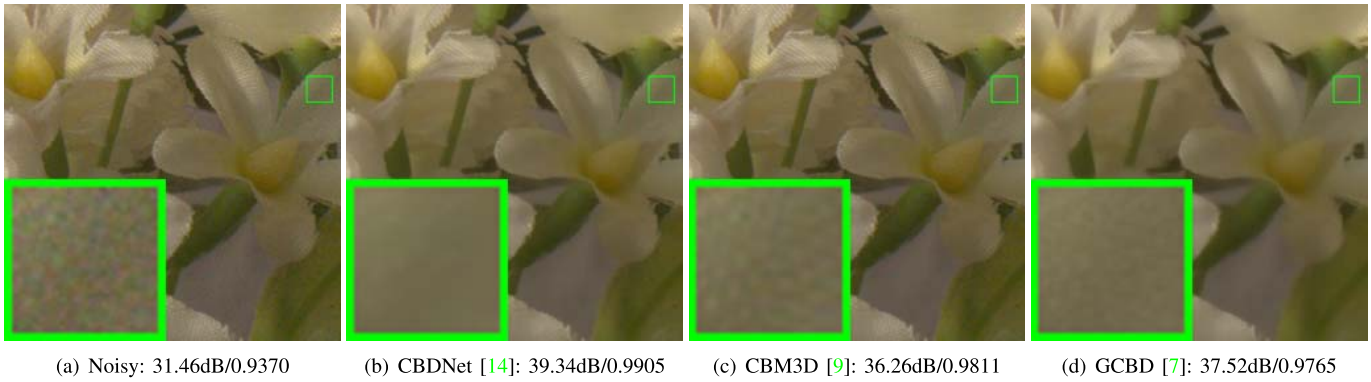
(a) Ground Truth (b) Noisy (36.25dB/0.9345) (c) CBM3D [9] (36.61dB/0.9669)

(d) NI [2] (37.58dB/0.9600) (e) DnCNN+ [41] (37.16dB/0.9389) (f) CBDNet [14] (36.58dB/0.9613)

(g) N2N [22] (36.99dB/0.9604) (h) DIP [23] (35.99dB/0.9529) (i) Blind ResNet+NAC (**37.88**dB/**0.9729**)

Fig. 6. **Denoised images and PSNR/SSIM results of " *5dmark3-iso3200-1*" in the *Cross-Channel* dataset [28] by different methods**. The best results are highlighted in **bold**.

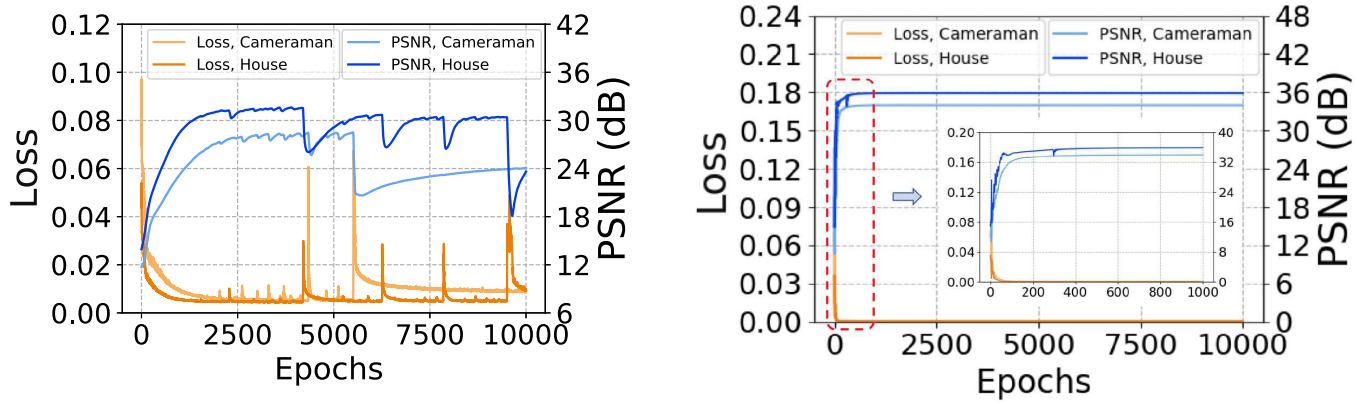
image $\mathbf{y}$ and its clean image $\mathbf{x}$ corrupted by AWGN noise or signal dependent Poisson noise. The experiments are performed on *Set12* dataset corrupted by AWGN or signal dependent Poisson noise. The noise deviations are in $\{5, 10, 15, 20, 25\}$. Figure 9 (a) shows comparisons of our ResNet+NAC and its "Oracle" networks on PSNR and SSIM. It can be seen that, the "Oracle" networks trained on the pair of noisy-clean images only perform slightly better than the original ResNet+NAC networks trained with the *simulated-observed* noisy image pairs $(\mathbf{z}, \mathbf{y})$. With our NAC strategy, the ResNet networks trained only with noisy test image achieves promising performance on weak noise.



(a) Noisy: 31.46dB/0.9370 (b) CBDNet [14]: 39.34dB/0.9905 (c) CBM3D [9]: 36.26dB/0.9811 (d) GCBD [7]: 37.52dB/0.9765

(e) NI [2]: 37.52dB/0.9868 (f) N2N [22]: 34.95dB/0.9621 (g) DnCNN+ [41]: 38.25dB/0.9888 (h) Blind ResNet+NAC: 38.34dB/0.9887

Fig. 7. **Denoised images and PSNR(dB)/SSIM by comparison methods** on "*0017_3*" in *DND* [29]. The "ground-truth" image is not released, but PSNR(dB)/SSIM results are publicly provided on https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb *DND* Benchmark.



(a) Curves of DIP [23] (b) Curves of our NAC

Fig. 8. **Training loss and PSNR (dB) curves** of DIP [23] (a) and our ResNet+NAC (b) networks w.r.t. the number of epochs, on the images of "Cameraman" and "House" from *Set12*.

TABLE VI

AVERAGE PSNR (dB) AND TIME (SECONDS) OF RESNET+NAC WITH DIFFERENT NUMBER OF EPOCHS ON *Set12* CORRUPTED BY AWGN NOISE ($\sigma = 15$)

# of Epochs	100	200	500	1000	5000
PSNR↑	31.80	32.79	33.77	34.24	34.28
SSIM↑	0.8714	0.9023	0.9189	0.9277	0.9280
Time↓	67.4	132.5	302.0	583.2	2815.6

4) Performance on Strong Noise: Our NAC strategy is based on the assumption of "weak noise". It is natural to wonder how well ResNet+NAC performs against strong noise. To answer this question, we compare the ResNet+NAC networks with BM3D [10] and DnCNN [41], on *Set12* corrupted by AWGN noise with $\sigma = 50$. The PSNR and SSIM results are plotted in Figure 9 (b). One can see that, our ResNet+NAC networks are limited in handling strong AWGN noise, when compared with BM3D [10] and DnCNN [41]. To study how the strong noise limits the performance of our NAC strategy, we perform experiments on the Set12 dataset with various noise levels of $\sigma = 30, 40, 50$, and provide more PSNR/SSIM results in Table VII. As can be seen, the performance gap between our ResNet+NAC and DnCNN becomes larger when the noise level σ is stronger. That is, our ResNet+NAC is

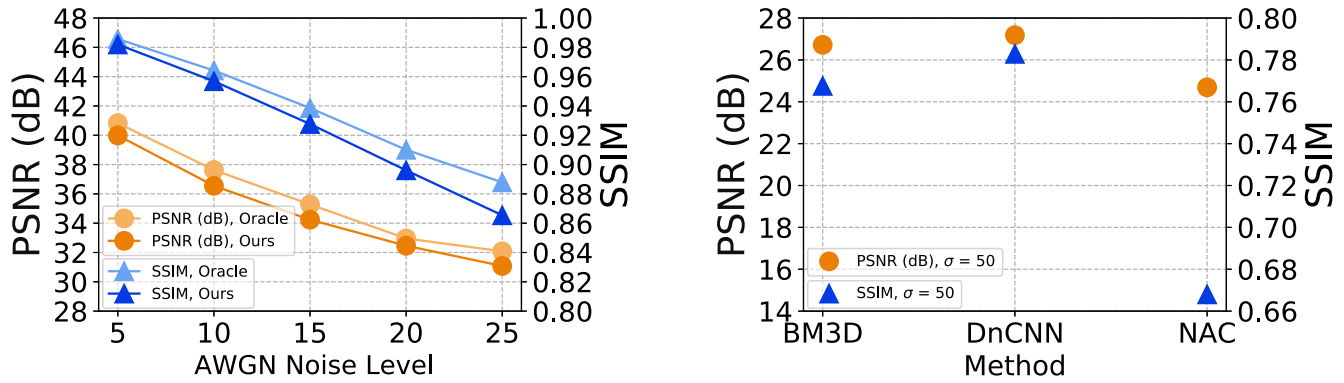


Fig. 9. **Comparisons of PSNR (dB) and SSIM results** on *Set12* (a) by our ResNet+NAC and its "Oracle" version for AWGN with $\sigma = 5, 10, 15, 20, 25$ and (b) by BM3D [10], DnCNN [41], and our ResNet+NAC for strong AWGN ($\sigma = 50$).

TABLE VII

AVERAGE PSNR (dB) AND SSIM [33] RESULTS OF BM3D [10], DNCNN [41] AND OUR RESNET+NAC ON THE SET12 DATASET CORRUPTED BY AWGN NOISE WITH $\sigma = 30, 40, 50$

Noise Level	$\sigma = 30$		$\sigma = 40$		$\sigma = 50$	
Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
BM3D [10]	29.14	0.8320	27.65	0.7944	26.72	0.7676
DnCNN [41]	29.52	0.8420	28.18	0.8094	27.18	0.7827
ResNet+NAC	29.91	0.8316	28.05	0.7722	24.69	0.6680

effective on "weak" noise and will degrade heavily when the noise becomes stronger.

VI. CONCLUSION

In this work, we proposed a "Noisy-As-Clean" (NAC) strategy for learning self-supervised image denoising networks. In our NAC, we trained an image-specific network by taking the corrupted image as the target, and adding to it the simulated noise to generate the doubly corrupted noisy input. The simulated noise is close to the observed noise in the noisy test image. This strategy can be seamlessly embedded into existing supervised denoising networks. We observed that *it is possible to learn a self-supervised network only with the corrupted image, approximating the optimal parameters of a supervised network learned with a pair of noisy and clean images.* Extensive experiments on synthetic and real-world benchmarks demonstrate that, the DnCNN [41] and ResNet (in Deep Image Prior [23]) trained with our NAC strategy achieved comparable or better performance on PSNR, SSIM, and visual quality, when compared to previous state-of-the-art image denoising methods, including supervised denoising networks. These results validate that our NAC strategy can learn effective image-specific priors and noise statistics only from the corrupted test image. As a potential future work, we will apply our NAC strategy on noisy document text images, since it is difficult to obtain their clean counterparts.

REFERENCES

[1] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1692–1700.

[2] N. ABSoft. *Neat Image*. Accessed: Aug. 20, 2020. [Online]. Available: https://ni.neatvideo.com/home

[3] J. Batson and L. Royer, "Noise2Self: Blind denoising by self-supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 524–533.

[4] P. Billingsley, *Probability and Measure* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 1995.

[5] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9446–9454.

[6] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.

[7] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.

[8] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. ICIP*, 2007, pp. 313–316.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[11] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao, "Conditional variational image deraining," *IEEE Trans. Image Process.*, vol. 29, pp. 6288–6301, 2020.

[12] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[13] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[14] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1712–1722.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[16] Y. Hou *et al.*, "NLH: A blind pixel-level non-local method for real-world image denoising," *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 5121–5135, Mar. 2020.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1–11.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[19] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2 Void–Learning denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2129–2137.

[20] S. Laine, T. Karras, J. Lehtinen, and T. Aila, "High-quality self-supervised deep image denoising," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 6970–6980.

[21] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3587–3596.

[22] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2971–2980.

[23] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454.

[24] Z. Liang, J. Xu, D. Zhang, Z. Cao, and L. Zhang, "A hybrid l1-l0 layer decomposition model for tone mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4758–4766.

[25] C. Liu, W. T. Freeman, R. Szeliski, and S. Bing Kang, "Noise estimation from a single image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 901–908.

[26] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1673–1682.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[28] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1683–1691,

[29] T. Plötz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1586–1595.

[30] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1087–1098.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 1–9.

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[34] J. Xu *et al.*, "STAR: A structure and texture aware retinex model," *IEEE Trans. Image Process.*, vol. 29, pp. 5022–5037, 2020.

[35] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, "Real-world noisy image denoising: A new benchmark," *CoRR*, vol. abs/1804.02603, pp. 1–13, Apr. 2018.

[36] J. Xu, D. Ren, L. Zhang, and D. Zhang, "Patch group based Bayesian learning for blind image denoising," in *Proc. Asian Conf. Comput. Vis. New Trends Image Restoration Enhancement Workshop*, 2016, pp. 79–95.

[37] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2996–3010, Jun. 2018.

[38] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 20–36.

[39] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1096–1104.

[40] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 244–252.

[41] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[42] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *Proc. CVPR*, Jun. 2011, pp. 977–984.

[43] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 479–486.

Jun Xu (Member, IEEE) received the B.Sc. degree in pure mathematics and the M.Sc. degree from the School of Mathematics Science, Nankai University, in 2011 and 2014, respectively, and the Ph.D. degree from the Hong Kong Polytechnic University, in 2018. He worked as a Research Scientist with the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. He is currently an Assistant Professor with the College of Computer Science, Nankai University.

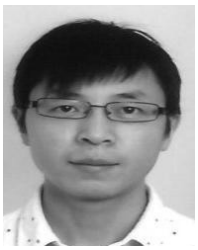
Yuan Huang received the B.Sc. degree in electronic engineering from the School of Electronic Information, Hangzhou Electronic University, and the M.Sc. degree from the School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications. She is currently pursuing the Ph.D. degree in computer vision with the School of Telecommunications, Xi'an Jiaotong University.

Ming-Ming Cheng (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University in 2012. Then he did two years research fellow with Prof. Philip Torr in Oxford. He is currently a Professor with Nankai University, leading the Media Computing Laboratory. His research interests include computer graphics, computer vision, and image processing. He received research awards, including the ACM China Rising Star Award, the IBM Global SUR Award, and the CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TIP.

Li Liu (Senior Member, IEEE) received the B.Eng. degree in electronic information engineering from Xi'an Jiaotong University, Xi'an, China, in 2011, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., in 2014. He is currently the Director of research with the Inception Institute of Artificial Intelligence. His current research interests include computer vision, machine learning, and data mining.

Fan Zhu received the M.Sc. degree (Hons.) in electrical engineering and the Ph.D. degree from the Visual Information Engineering Group, Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K, in 2011 and 2015, respectively. He was with New York University Abu Dhabi as a Postdoctoral Researcher from 2015 to 2017. He is currently with the Inception Institute of Artificial Intelligence. His research interests include scene understanding, 3D shape representation, generation, and network compression.

Zhou Xu received the dual Ph.D. degree from the School of Computer Science, Wuhan University, China, and the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He is currently an Assistant Professor with the School of Big Data and Software Engineering, Chongqing University, Chongqing, China. His research interests include feature engineering and data mining.



Ling Shao (Senior Member, IEEE) is currently the Executive Vice President and a Provost of the Mohamed bin Zayed University of Artificial Intelligence. He is also the CEO and the Chief Scientist of the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. His research interests include computer vision, machine learning, and medical imaging. He is a Fellow of IAPR, IET, and BCS.