Noisy-As-Clean: Learning Unsupervised Denoising from the Corrupted Image

Jun Xu[1,#,*], Yuan Huang[2,#], Ming-Ming Cheng[1], Li Liu[3], Fan Zhu[3], Xingsong Hou[2], Ling Shao[3]
[1]College of Computer Science, Nankai University, Tianjin, China
[2]School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
[3]Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE

Abstract

Recently, 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 also proposed to use external noisy images for training. However, for an unseen test image, these denoising 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 that 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 bridge this gap, in this work, we propose a novel "Noisy-As-Clean" (NAC) strategy of training unsupervised denoising networks. In NAC, the corrupted test image is taken as the "clean" target, while the inputs are simulated images consisting of the test image and similar corruptions. A simple and useful observation on our NAC is: ***as long as the noise is weak, it is feasible to learn an unsupervised network only with the corrupted image, approximating the optimal parameters of a supervised network learned with pairs of noisy and clean images****. Experiments show that the unsupervised networks trained with our NAC strategy outperform previous networks, including supervised ones, on synthetic and realistic noise removal.*

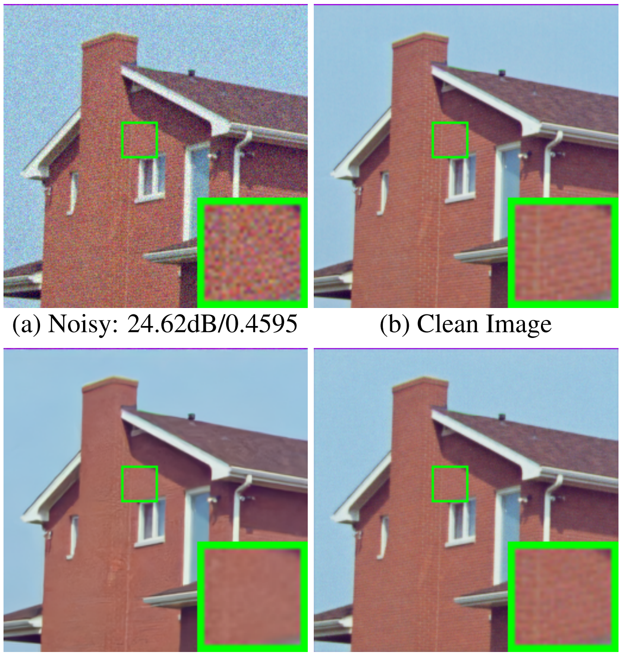

(a) Noisy: 24.62dB/0.4595
(b) Clean Image
(c) CDnCNN [44]: 34.23dB/0.8695
(d) CDnCNN+NAC: **35.80**dB/**0.9116**

Figure 1. Denoised images and PSNR/SSIM results of CDnCNN [44] (c) and CDnCNN trained by our NAC strategy ("CDnCNN+NAC") (d) on the color image *House* corrupted by AWGN noise ($\sigma = 15$).

1. 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. AWGN serves as a perfect test bed for supervised methods in the deep learning era [18, 19, 22, 38, 39]. Numerous supervised networks [9, 17, 25, 29, 31, 35, 40, 42, 44] 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 [25, 29, 31, 35, 40, 44], a 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 [4, 15, 28, 33, 34]. 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 test photograph. Recently, several unsupervised networks [6, 10, 23, 24, 26, 27] have

[*] Jun Xu (nankaimathxujun@gmail.com) is the corresponding author. The first two authors contribute equally.

been proposed to remove the dependence on clean training images. However, when processing a corrupted test image, these networks are still subjected to the domain gap on either image priors or noise statistics between the external training images and the test one. Two interesting works are Noise2Noise [26] and Deep Image Prior (DIP) [27], which succeed on the zero-mean noise. But the realistic noise on real photographs is usually not zero-mean [4, 33, 34].

In this paper, we propose a "Noisy-As-Clean" (NAC) strategy to alleviate the *domain gap problem*. In our NAC, the *noisy* test image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ is directly taken *as* the "*clean*" target to train an image-specific network. Thus, the domain gap on image priors are largely bridged by our NAC. To reduce the gap on noise statistics, for the noisy test image $\mathbf{y}$ as target, we take as the input of our NAC a *simulated* noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$ consisting of the corrupted test 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, in the training stage, our NAC network learns to remove the *simulated* noise $\mathbf{n}_s$ from $\mathbf{z}$, and thus is able to remove the *observed* noise $\mathbf{n}_o$ from the noisy test image $\mathbf{y}$ in the testing stage.

A simple and useful observation about our NAC strategy is: *as long as the corrupted noise is "weak", it is feasible to train an unsupervised denoising network only with the corrupted test image, and the learned parameters are very close to those of a supervised network trained with a pair of noisy and clean images.* Though being very simple, our NAC strategy is very effective for image denoising. In Figure 1, we compare the denoised images by the vanilla CDnCNN [44] and the CDnCNN trained with our NAC (CDnCNN+NAC), on the image "House" corrupted by AWGN ($\sigma = 15$). We observe that the "CDnCNN+NAC" achieves better visual quality and higher PSNR/SSIM results than CDnCNN [44], which is trained on plenty of noisy and clean image pairs. Experiments on diverse synthetic and real-world benchmarks demonstrate that, when trained with our NAC strategy, an unpolished ResNet [18] outperforms previous supervised denoising networks on removing "weak" noise. Our work reveals that, *when the noise is "weak"*, an unsupervised network trained with only the corrupted test image can obtain better denoising performance than the supervised ones.

2. Related Work

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. Numerous methods [9, 17, 25, 29, 31, 35, 40, 42, 44] 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 noisy photographs [1, 3, 4, 34].

Type	Method	Year’Pub.	Image Prior	Noise Stat.
S.	DnCNN [44]	17’TIP	Ext.	✓
	CBDNet [17]	19’CVPR	Ext.	✓
U.	N2N [26]	18’ICML	Ext.	✓
	DIP [27]	18’CVPR	Int.	
	N2S [6]	19’ICML	Ext.	
	N2V [23]	19’CVPR	Ext.	
	SS [24]	19’NeurIPS	Ext.	
	NAC (Ours)	20’Submit	Int.	✓

Table 1. **Summary of representative networks for image denoising**. **S.**: Supervised networks. **U.**: Unsupervised networks. **N2N**: Noise2Noise [26]. **DIP**: Deep Image Prior [27]. **N2S**: Noise2Self [6]. **N2V**: Noise2Void [23]. **SS**: Self-Supervised [24]. **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.

Unsupervised denoising networks are developed to remove the need on plenty of clean images. Along this direction, Noise2Noise (N2N) [26] 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) [23] 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) [6], N2V [23] significantly degrades the training efficiency and denoising performance at test time. Recently, Deep Image Prior (DIP) [27] 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.

Internal and external image priors are widely used for diverse image restoration tasks [14, 36, 43, 45, 46]. Internal priors are directly learned from the input test image itself, while the external ones are learned on external images (as long as not the test one). The internal priors are adaptive to its image contents, but somewhat affected by the corruptions [14, 45]. By contrast, the external priors are effective for restoring images with general contents, but may not be optimal for specific test image [11, 14, 36, 37, 43, 46].

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 [4, 34], which is usually modeled as mixed Poisson and Gaussian distribution [15]. The Poisson component mainly comes from the irregular photons hitting the sensor [28], while Gaussian noise is majorly produced by dark current [33]. Though performing well on the synthetic noise being trained with, supervised denoisers [9, 17, 25, 29, 31, 40, 42, 44] still suffer from the *domain gap problem* when processing the real noisy photographs.

In Table 1, we summarize several recently published denoising networks [6, 8, 23, 26, 27, 29, 31, 35, 40, 44] from the aspects of supervised and unsupervised 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 priors and noise statistics directly from the corrupted test image.

3. Theoretical Background of "Noisy-As-Clean" Strategy

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

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

Assume 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

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

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

Different with the "zero-mean" assumption in [23, 26], here we study a practical assumption on noise statistics, i.e., *the expectation* $\mathbb{E}[\mathbf{x}]$ *and variance* $\text{Var}[\mathbf{x}]$ *of signal intensity are much stronger than those of noise* $\mathbb{E}[\mathbf{n}_o]$ *and* $\text{Var}[\mathbf{n}_o]$ (such that they are negligible but not necessarily zero):

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

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 $\text{Var}[\mathbf{n}_s] \approx \text{Var}[\mathbf{n}_o]$. Then we have

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

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* [7], 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^*:

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

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

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

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

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

Our observation is very simple and useful: *as long as the noise is weak, the optimal parameters of unsupervised 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* $\{(\mathbf{y}_i, \mathbf{x}_i)\}$.

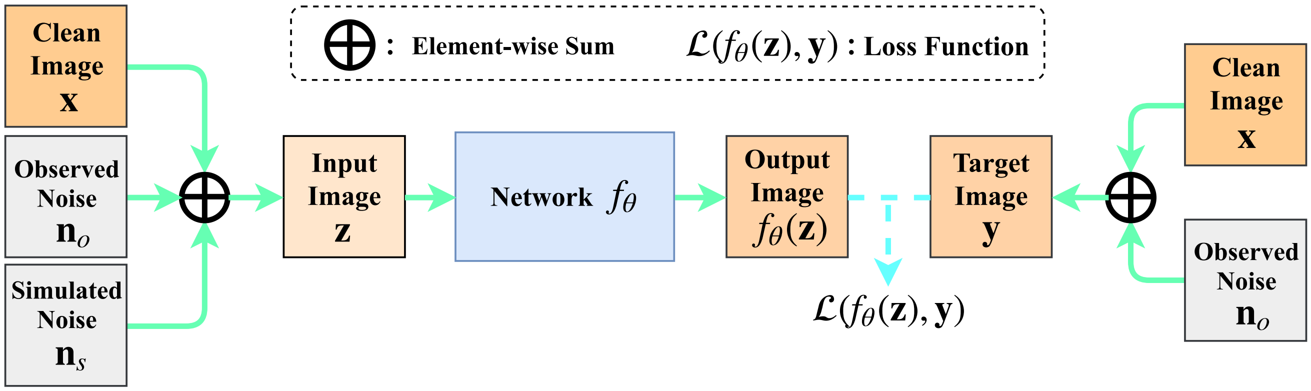

Clean Image $\mathbf{x}$
Observed Noise $\mathbf{n}_o$
Simulated Noise $\mathbf{n}_s$
Input Image $\mathbf{z}$
Network f_θ
Output Image $f_\theta(\mathbf{z})$
Target Image $\mathbf{y}$
$\mathcal{L}(f_\theta(\mathbf{z}), \mathbf{y})$
⊕ : Element-wise Sum
$\mathcal{L}(f_\theta(\mathbf{z}), \mathbf{y})$: Loss Function
Clean Image $\mathbf{x}$
Observed Noise $\mathbf{n}_o$

Figure 2. **Proposed "Noisy-As-Clean" strategy for training unsupervised image denoising networks**. In this 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.

Consistency of noise statistics. Since our contexts are the real-world scenarios, the noise can be modeled by mixed Poisson and Gaussian distribution [15]. 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.,

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

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 "$\approx$" indicates that the observed noise $\mathbf{n}_o$ is negligible. Thus we have

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

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 can be seen by the experiments (§5), this property makes our "Noisy-As-Clean" strategy consistently effective on different noise removal tasks.

4. Learning "Noisy-As-Clean" Networks for Unsupervised Image Denoising

With our statistical analysis, here we propose to learn unsupervised networks with our "Noisy-As-Clean" (NAC) strategy for image denoising. Note that we only need the *observed* noisy image $\mathbf{y}$ to generate noisy image pairs $\{(\mathbf{z}, \mathbf{y})\}$ with *simulated* noise $\mathbf{n}_s$. Our idea is illustrated in Figure 2.

Training NAC networks. For real-world images captured by camera sensors, one can hardly distinguish the realistic noise from the signal. Our observation is that 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 $\mathbf{x}$. We can observe that, 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 easily be removed. Note that we *do not* use the clean image $\mathbf{x}$ as "ground-truth" in training our NAC networks.

Training blind denoising. Most of existing supervised denoising networks train a specific model to process a fixed noise pattern [8, 9, 29, 33, 40]. To tackle the 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, our NAC networks should have the ability to blindly remove the unknown noise from real photographs. Inspired by [17, 44], we propose to train a blind version of our NAC networks by using the AWGN noise within a range of levels (e.g., $[0; 55]$) for removing unknown AWGN noise. We also use the mixed AWGN and Poisson noise (both within a range of intensities) for removing the realistic noise. More details will be introduced in §5.2.

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

Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$		$\sigma = 25$	
Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
BM3D [13]	38.07	0.9580	34.40	0.9234	32.38	0.8957	31.00	0.8717	29.97	0.8503
DnCNN [44]	38.76	0.9633	34.78	0.9270	32.86	0.9027	31.45	0.8799	30.43	0.8617
N2N [26]	**39.72**	0.9665	36.18	0.9446	33.99	0.9149	32.10	0.8788	30.72	0.8446
DIP [27]	32.49	0.9344	31.49	0.9299	29.59	0.8636	27.67	0.8531	25.82	0.7723
N2V [23]	27.06	0.8174	26.79	0.7859	26.12	0.7468	25.89	0.7405	25.01	0.6564
NAC	**39.99**	**0.9820**	**36.55**	**0.9569**	**34.24**	**0.9277**	**32.46**	**0.8961**	**31.08**	**0.8654**
Blind-NAC	38.48	**0.9805**	**36.65**	**0.9564**	**34.77**	**0.9275**	**33.13**	**0.9024**	**31.78**	**0.8802**

Table 2. **Average PSNR (dB) and SSIM [41] results of different methods on *Set12* dataset** corrupted by AWGN noise. The best and second best results are highlighted in **red** and **blue**, 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 [13]	37.59	0.9640	33.32	0.9163	31.07	0.8720	29.62	0.8342	28.57	0.8017
DnCNN [44]	38.07	**0.9695**	33.88	**0.9270**	31.73	0.8706	30.27	0.8563	**29.23**	**0.8278**
N2N [26]	**38.58**	0.9627	34.07	0.9200	31.81	0.8770	30.14	0.8550	28.67	0.8123
DIP [27]	29.74	0.8435	28.16	0.8310	27.07	0.7867	25.80	0.7205	24.63	0.6680
N2V [23]	26.70	0.7915	26.39	0.7621	25.77	0.7126	25.41	0.6678	24.83	0.6305
NAC	**39.00**	**0.9707**	**34.60**	**0.9324**	**32.13**	**0.8942**	**30.47**	**0.8636**	28.96	0.8185
Blind-NAC	38.26	0.9605	**34.26**	0.9266	**32.06**	**0.8919**	**30.50**	**0.8609**	**29.33**	**0.8327**

Table 3. **Average PSNR (dB) and SSIM [41] results of different methods on *BSD68* dataset** corrupted by AWGN noise. The best and second best results are highlighted in **red** and **blue**, respectively.

which the objective metrics, e.g., PSNR and SSIM [41], can be computed with the clean image $\mathbf{x}$.

Implementation details. We employ the ResNet-20 network used in [27] as the backbone network, which includes 10 residual blocks. Each block contains two convolutional layers followed by a Batch Normalization (BN) [20]. The Rectified Linear Units (ReLU) activation operator [32] is used after the first BN. Its parameters are randomly initialized *without being pretrained*. The optimizer is Adam [21] with default parameters. The learning rate is fixed at 0.001 in all experiments. We use the ℓ_2 loss function. The network is trained in 1000 epochs for each test image. For data augmentation, we employ 4 rotations $\{0°, 90°, 180°, 270°\}$ combined with 2 mirror (vertical and horizontal) reflections, resulting in totally 8 transformations. We implement our ResNet based NAC networks in PyTorch [2].

5. 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 $\mathbf{z}$ (with data augmentation) 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 [41], 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 §5.1, continue to perform blind image denoising in §5.2, and finally tackle the realistic noise in §5.3. In §5.4, we conduct comprehensive ablation studies to gain deeper insights into the proposed NAC strategy.

5.1. Synthetic Noise Removal With Known Noise

We evaluate the proposed NAC networks on images corrupted by synthetic noise such as AWGN. More experimental results on signal dependent Poisson noise and mixed Poisson-AWGN noise are provided in the *Supplementary File*.

Training NAC 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 $\mathbf{y}$.

Comparison methods. We compare our NAC networks with state-of-the-art image denoising methods [13, 26, 27, 30, 44]. On AWGN noise, we compare with BM3D [13], DnCNN [44], Noise2Noise (N2N) [26], Deep Image Prior (DIP) [27], and Noise2Void (N2V) [23].

Test datasets. We evaluate the comparison methods on the *Set12* and *BSD68* datasets, which are widely tested by supervised denoising networks [29, 31, 40, 44] and previous methods [13, 16, 43, 46]. The *Set12* dataset contains 12 images of sizes 512×512 or 256×256, while the *BSD68* dataset contains 68 images of different sizes.

Results on AWGN noise. We test AWGN with noise deviation (noise level) of $\sigma \in \{5, 10, 15, 20, 25\}$, i.e., the *observed* noise $\mathbf{n}_o$ is AWGN with standard deviation (std) of

Dataset	Type	Traditional Methods		Supervised Networks		Unsupervised Networks			
	Method	**CBM3D** [12]	**NI** [5]	**DnCNN+** [44]	**CBDNet** [17]	**GCBD** [10]	**N2N** [26]	**DIP** [27]	**NAC**
CC [33]	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 [34]	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

Table 4. **Average PSNR (dB) and SSIM [41] of different methods** on the *CC* dataset [33] and the *DND* dataset [34]. The best results are highlighted in **bold**. "NA" means "Not Available" due to unavailable code (GCBD on *CC* [33]) or difficult experiments (DIP on *DND* [34]).

σ. Since AWGN noise is signal independent, the *simulated* noise $\mathbf{n}_s$ is set with the same σ as that of $\mathbf{n}_o$. The comparison results are listed in Tables 2 and 3. It can be seen that, the network trained with the proposed NAC networks achieve much better performance on PSNR and SSIM [41] than BM3D [13] and DnCNN [44], two previous leading image denoising methods. Note that DnCNN are supervised networks trained on clean and synthetic noisy image pairs. Our NAC networks outperform the other unsupervised networks N2N [26], DIP [27], and N2V [23] by a significant margin on PSNR and SSIM [41].

5.2. Synthetic Noise Removal With Unknown Noise

To deal with unknown noise, we propose to train a blind version of our NAC networks for removing unknown noise. Here, we test our NAC networks on AWGN noise with unknown noise deviation. We use the same training strategy, comparison methods, and test datasets as in §5.1.

Training blind NAC networks. For each test image, we train a NAC network corrupted by AWGN 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 NAC network on the test image with different deviations. Our NAC networks trained on AWGN with unknown noise levels is termed as "Blind-NAC".

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 NAC network is directly utilized to denoise the test image without estimating its deviation. The results are also listed in Tables 2 and 3. We observe that, our Blind-NAC networks trained on AWGN noise with unknown levels can achieve even better PSNR and SSIM [41] results than our NAC networks trained on specific noise levels. Note that on *BSD68*, our Blind-NAC networks achieve better performance than DnCNN [44], which achieves higher PSNR and SSIM results than our NAC networks. This demonstrate the effectiveness of our 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 unknown and complex.

5.3. 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 [15, 17]. Hence, for each *observed* noisy image $\mathbf{y}$, we generate the *simulated* noise $\mathbf{n}_s$ by sampling the $\mathbf{y}$-dependent Poisson part and the independent AWGN noise.

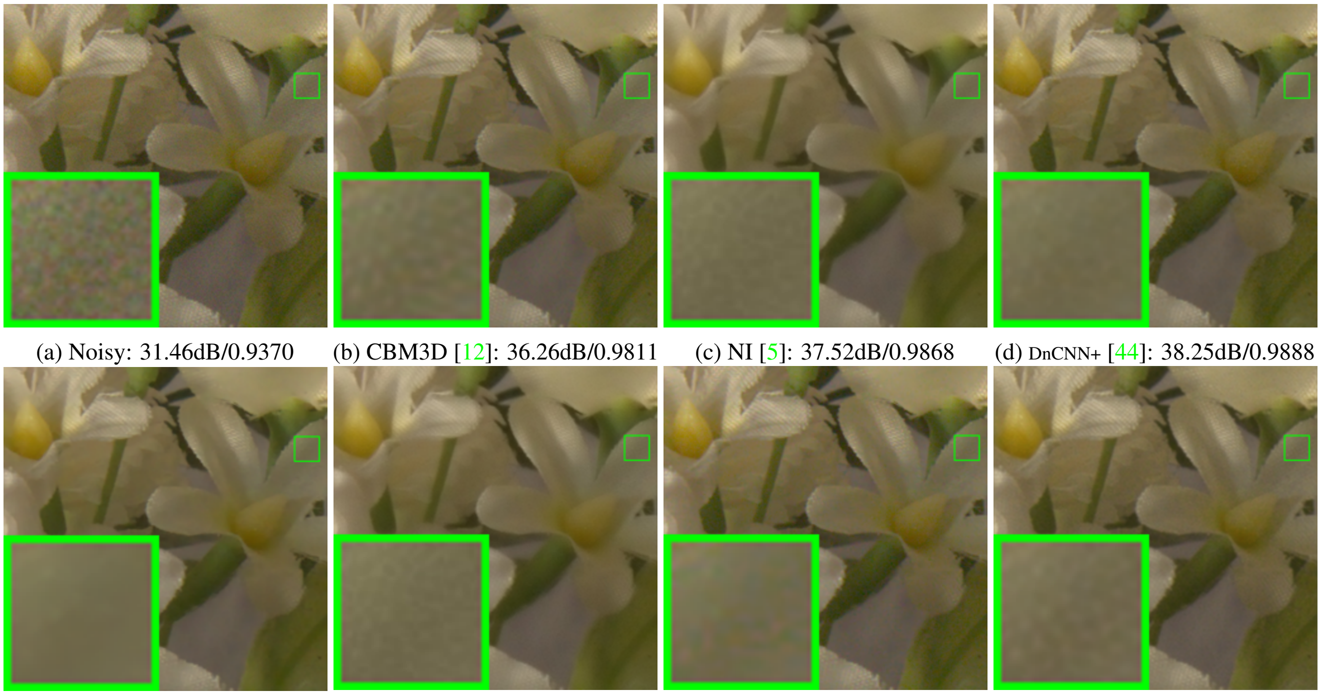
Training blind NAC networks is also performed for each test image, i.e., the *observed* noisy image $\mathbf{y}$. In real-world scenarios, each *observed* noisy image $\mathbf{y}$ is corrupted without knowing the specific noise statistics of the *observed* noise $\mathbf{n}_o$. Therefore, the *simulated* noise $\mathbf{n}_s$ is directly estimated on $\mathbf{y}$ as mixed $\mathbf{y}$-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 $0 < \sigma \leq 25$.

Comparison methods. We compare with state-of-the-art methods on real-world image denoising, including CBM3D [12], the commercial software Neat Image [5], two supervised networks DnCNN+ [44] and CBDNet [17], and three unsupervised networks GCBD [10], Noise2Noise [26], and DIP [27]. Note that DnCNN+ [44] and CBDNet [17] are two state-of-the-art supervised networks for real-world image denoising, and DnCNN+ is an improved extension of DnCNN [44] with much better performance (the authors of DnCNN+ provide us the models/results of DnCNN+).

Test datasets. We evaluate the comparison methods on the *Cross-Channel* (*CC*) dataset [33] and *DND* dataset [34].

The *CC* dataset [33] includes noisy images of 11 static scenes captured by Canon 5D Mark 3, Nikon D600, and Nikon D800 cameras. The real-world noisy images are collected under a highly controlled indoor environment. Each scene is shot 500 times using the same camera and settings. The average of the 500 shots is taken as the "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 [34] 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



(a) Noisy: 31.46dB/0.9370 (b) CBM3D [12]: 36.26dB/0.9811 (c) NI [5]: 37.52dB/0.9868 (d) DnCNN+ [44]: 38.25dB/0.9888

(e) CBDNet [17]: 39.34dB/0.9905 (f) GCBD [10]: 37.52dB/0.9765 (g) N2N [26]: 34.95dB/0.9621 (h) NAC: 38.34dB/0.9887

Figure 3. **Denoised images and PSNR(dB)/SSIM by comparison methods** on "*0017_3*" in *DND* [34]. The "ground-truth" image is not released, but PSNR(dB)/SSIM results are publicly provided on *DND* Benchmark.

shorter exposure times, while the "ground 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 performance on PSNR and SSIM by submitting the denoised images to the *DND*'s Website.

Results on PSNR and SSIM. The comparisons on average PSNR and SSIM results are listed in Table 4. As can be seen, the proposed NAC networks achieve better performance than all previous denoising methods, including the CBM3D [12], the supervised networks DnCNN+ [44] and CBDNet [17], and the unsupervised networks GCBD [10], N2N [26], and DIP [27]. This demonstrates that the proposed NAC networks can indeed handle the complex, unknown, and realistic noise, and achieve better performance than supervised networks such as DnCNN+ [44] and CBDNet [17].

Qualitative results. In Figure 3, we show the denoised images of our NAC network and the comparison methods on the image "*0017_3*" from the *DND* dataset. We observe that, our unsupervised NAC networks are very effective on removing realistic noise from the real photograph. Besides, our NAC networks achieve competitive PSNR and SSIM results when compared with the other methods, including the supervised ones such as DnCNN+ [44] and CBDNet [17].

Speed. The work most similar to ours is Deep Image Prior (DIP) [27], 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 NAC network needs 583.2 seconds (on an NVIDIA Titan X GPU).

5.4. Ablation Study

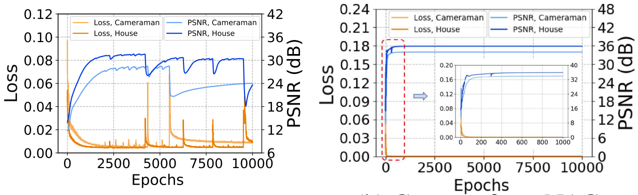
To further study our NAC strategy, we conduct detailed examination of our NAC networks on image denoising.

1) Generality of our NAC strategy. To evaluate the generality of the proposed NAC strategy, we apply it on the DnCNN [44] network and denote the resulting network as "DnCNN-NAC". We train DnCNN with our NAC strategy (DnCNN-NAC), and the comparison results with DnCNN are listed in Tab. 5. One can see that DnCNN-NAC achieves better PSNR results than that of the original DnCNN when $\sigma = 5, 10, 15$ (but worse when $\sigma = 20, 25$). Note that the original DnCNN network is trained offline on the *BSD400* dataset, while here the DnCNN-NAC network is trained online for each specific test image.

σ	5	10	15	20	25
DnCNN [44]	38.76	34.78	32.86	31.45	30.43
DnCNN-NAC	43.18	37.16	33.65	31.16	29.23

Table 5. **PSNR (dB) results of DnCNN and DnCNN-NAC** on *Set12* corrupted by AWGN noise with different σ.

2) Differences from DIP [27]. Though the basic network in our work is the ResNet used in DIP [27], our NAC network is essentially different from DIP on at least two aspects. First, our NAC is a novel strategy for unsupervised learning of *adaptive network parameters* for the degraded image, while



(a) Curves of DIP [27] (b) Curves of our NAC

Figure 4. **Training loss and PSNR (dB) curves** of DIP [27] (a) and our NAC (b) networks w.r.t. the number of epochs, on the images of "Cameraman" and "House" from *Set12*.

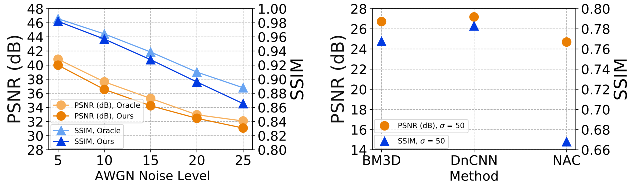
DIP aims to investigate *adaptive network structure* without learning the parameters. Second, our NAC learns a mapping from the synthetic noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$ to the noisy image $\mathbf{y}$, which approximates the mapping from the noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}_o$ 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 NAC achieves more robust (and better) denoising performance than DIP on diverse images. In Figure 4, we plot the curves of training loss and test PSNR of DIP (a) and 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 NAC can stably achieve better denoising results within 1000 epochs.

3) Influence on the number of residual blocks and epochs. Our backbone network is the ResNet [27] with 10 residual blocks trained in 1000 epochs. Now we study how the number of residual blocks and epochs influence the performance of NAC on image denoising. The experiments are performed on the *Set12* dataset corrupted by AWGN noise ($\sigma = 15$). From Table 6, we observe that, with more residual blocks, the NAC networks the NAC networks can achieve better PSNR and SSIM [41] 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 [27]. Then we study how the number of epochs influence the performance of NAC on image denoising. From Table 7, one can see that on the *Set12* dataset corrupted by AWGN noise ($\sigma = 15$), with more training epochs, our NAC networks achieve better PSNR and SSIM results, but with longer processing time.

# of Blocks	1	2	5	10	15
PSNR↑	33.58	33.85	34.14	34.24	34.28
SSIM↑	0.9161	0.9226	0.9272	0.9277	0.9272

Table 6. **Average PSNR (dB)/SSIM of NAC with different number of blocks** on *Set12* corrupted by AWGN noise ($\sigma = 15$).

4) Comparison with Oracle. We also study the "Oracle" performance of our NAC networks. In "Oracle", we train our NAC networks on the pair of *observed* noisy 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 5 (a) shows comparisons of our 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 NAC networks trained with the *simulated-observed* noisy image pairs $(\mathbf{z}, \mathbf{y})$. With our NAC strategy, the NAC networks trained only with the noisy test image can achieve similar denoising performance on the weak noise.



(a) (b)

Figure 5. **Comparisons of PSNR (dB) and SSIM results** on *Set12* (a) by our NAC networks and its "Oracle" version for AWGN with $\sigma = 5, 10, 15, 20, 25$ and (b) by BM3D [13], DnCNN [44], and our NAC networks for strong AWGN ($\sigma = 50$).

5) Performance on strong noise. Our NAC strategy is based on the assumption of "weak noise". It is natural to wonder how well NAC performs against strong noise. To answer this question, we compare the NAC networks with BM3D [13] and DnCNN [44], on *Set12* corrupted by AWGN noise with $\sigma = 50$. The PSNR and SSIM results are plotted in Figure 5 (b). One can see that, our NAC networks are limited in handling strong AWGN noise, when compared with BM3D [13] and DnCNN [44].

# of Epochs	100	200	500	1000	5000
PSNR↑	31.80	32.79	33.77	34.24	34.32
SSIM↑	0.8714	0.9023	0.9189	0.9277	0.9280
Time↓	67.4	132.5	302.0	583.2	2815.6

Table 7. **Average PSNR (dB) and time (s) of NAC with different number of epochs** on *Set12* corrupted by AWGN noise ($\sigma = 15$).

6. Conclusion

In this work, we proposed a "Noisy-As-Clean" (NAC) strategy for learning unsupervised image denoising networks. In our NAC, we trained an image-specific network by taking the noisy test image as the target, and adding to it the simulated noise to generate the *simulated* 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 provided a simple and useful observation: *it is possible to learn an unsupervised network only with the noisy image, approximating the optimal parameters of a supervised network learned with pairs of noisy and clean images.* Extensive experiments on benchmark datasets demonstrate that, the networks trained with our NAC strategy achieved better comparable performance on PSNR, SSIM, and visual quality, when compared to previous state-of-the-art image denoising methods, including several supervised learning denoising networks. These results validate that our NAC strategy can learn image-specific priors and noise statistics only from the corrupted test image.

References

[1] Darmstadt Noise Dataset Benchmark. https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb. Accessed: 2019-05-23. 2

[2] PyTorch. https://pytorch.org/. Accessed: 2019-05-23. 5

[3] Smartphone Image Denoising Dataset Benchmark. https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php. Accessed: 2019-05-23. 2

[4] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, June 2018. 1, 2

[5] Neatlab ABSoft. Neat Image. https://ni.neatvideo.com/home. 6, 7

[6] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *ICML*, volume 97, pages 524–533. PMLR, 2019. 1, 2, 3

[7] Patrick Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995. 3

[8] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, pages 9446–9454, 2019. 3, 4

[9] Harold Christopher Burger, Christian J. Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR*, pages 2392–2399, 2012. 1, 2, 3, 4

[10] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, pages 3155–3164, 2018. 1, 6, 7

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

[12] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *ICIP*, pages 313–316. IEEE, 2007. 6, 7

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 5, 6, 8

[14] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 2

[15] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, Oct 2008. 1, 2, 4, 6

[16] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, 121(2):183–208, 2017. 5

[17] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 1, 2, 3, 4, 6, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2

[19] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 1

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[23] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void-learning denoising from single noisy images. In *CVPR*, 2019. 1, 2, 3, 5, 6

[24] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *NeurIPS*, 2019. 1, 2

[25] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, pages 3587–3596, 2017. 1, 2, 3

[26] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, pages 2971–2980, 2018. 1, 2, 3, 5, 6, 7

[27] Victor Lempitsky, Dmitry Ulya Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018. 1, 2, 3, 5, 6, 7, 8

[28] Ce Liu, William T. Freeman, Richard Szeliski, and Sing Bing Kang. Noise estimation from a single image. *CVPR*, 1:901–908, 2006. 1, 3

[29] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S. Huang. Non-local recurrent network for image restoration. In *NeurIPS*, pages 1673–1682. 2018. 1, 2, 3, 4, 5

[30] Markku Makitalo and Alessandro Foi. Optimal inversion of the anscombe transformation in low-count poisson image denoising. *IEEE Transactions on Image Processing*, 20(1):99–109, 2011. 5

[31] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. In *NIPS*, 2016. 1, 2, 3, 5

[32] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. 5

[33] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. *In CVPR*, pages 1683–1691, 2016. 1, 2, 3, 4, 6

[34] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 1, 2, 6, 7

[35] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018. 1, 2, 3

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

[37] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, June 2014. 2

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1

[40] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 1, 2, 3, 4, 5

[41] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5, 6, 8

[42] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, pages 341–349, 2012. 1, 2, 3

[43] Jun Xu, Wangmeng Zuo, Lei Zhang, David Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *ICCV*, pages 244–252, 2015. 2, 5

[44] 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*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[45] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR*, 2011. 2

[46] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. 2, 5