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

Jun Xu^{1,*}, Yuan Huang^{1,2,*}, Li Liu¹, Fan Zhu¹, Xingsong Hou², Ling Shao¹

¹Inception Institute of Artificial Intelligence, United Arab Emirates

²Xi'an Jiaotong University, China

nankaimathxujun@gmail.com

Abstract

In the past few years, supervised networks have achieved promising performance on image denoising. These methods learn image priors and synthetic noise statistics from plenty pairs of noisy and clean images. Recently, several unsupervised denoising networks are proposed only using external noisy images for training. However, the networks learned from external data inherently suffer from the *domain gap dilemma*, i.e., the image priors and noise statistics are very different between the training data and the corrupted test images. This dilemma becomes more clear when dealing with the signal dependent realistic noise in real photographs. In this work, we provide a statistically useful conclusion: *it is possible 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.* This is achieved by proposing a “Noisy-As-Clean” strategy: taking the corrupted image as “clean” target and the simulated noisy images (based on the corrupted image) as inputs. Extensive experiments show that the unsupervised denoising networks learned with our “Noisy-As-Clean” strategy surprisingly outperforms previous supervised networks on removing several typical synthetic noise and realistic noise. The code will be publicly released.

1 Introduction

Image denoising is an ill-posed inverse problem aiming to recover a *clean* image \mathbf{x} from the *observed* noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where \mathbf{n} is the 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 neural networks (DNNs) era [1–5]. To achieve success on image denoising, existing supervised networks [6–13] explore to learn the image priors and noise statistics on plenty pairs of clean and corrupted images, and achieving promising performance on recovering the images with similar priors and noise statistics (e.g., AWGN).

While advancing the problem of synthetic AWGN noise removal, researchers recently have shifted their interests to the real-world scenarios, where the noise is more complex than AWGN [14]. Since the realistic noise is signal dependent [14–16], its statistics in a test real photograph is very different from the simulated ones in the training images. That is, the supervised networks unavoidably suffer from a *domain gap dilemma*: both the training images and simulated noise are very different from the test ones in real scenarios. Therefore, training supervised networks by simulating the realistic noise (e.g., as mixed Poisson and Gaussian) is still problematic [17]. Supervised networks would stagnate on current stage unless this *domain gap dilemma* is solved fundamentally. Recently, several

*Corresponding author: Jun Xu (nankaimathxujun@gmail.com)

unsupervised networks [18–21] have been proposed to consider the problem in real-world scenarios, motivated by the largely impossible acquisition of clean training targets. Two interesting work along this line are Noise2Noise [18] and Deep Image Prior [19]. These methods are effective based on the assumption that the noise is zero-mean distributed [18, 19]. However, the *signal dependent Poisson noise dominating real photographs is not necessarily zero-mean*. Besides, these unsupervised networks did not solve the *domain gap dilemma*, which will be explained in §2.

In this work, we provide a simple and useful conclusion: it is statistically possible to learn an unsupervised network for image denoising only with the *observed* noisy image, and the learned optimal parameters are very close to those of a *supervised* network, which is trained with pairs of *observed* noisy images and their *clean* counterparts. This is achieved by learning networks in a proposed “Noisy-As-Clean” strategy: we take the observed *noisy* image as the “*clean*” target and take the *simulated* noisy images as inputs, each of which is produced by adding to the observed *noisy* image *simulated* noise, which is statistically the same (or similar) to the *observed* noise.

The *domain gap dilemma* in supervised denoising networks is naturally tackled with the “Noisy-As-Clean” strategy. On one hand, the *observed* noisy image we test is taken as the target image for learning, thus the gap in image priors is bridged. On the other, since the *simulated* noise is statistically close to the *observed* noise, the gap in noise statistics is largely decreased. Though being very simple, the proposed “Noisy-As-Clean” strategy is very effective for image denoising. For example, when trained with the “Noisy-As-Clean” strategy, an unpolished ResNet [4] can surprisingly outperform, by a large margin, the supervised denoising networks like DnCNN [11] and CBDNet [17], on removing AWGN and realistic noise. This work contributes to the practical image denoising problem by minimizing the requirements, i.e., only the corrupted image, on availability of training data.

2 Related Work

Existing image denoising networks can be roughly divided into supervised and unsupervised categories. In Table 1, we compare recently developed networks [8, 9, 11–13, 18–22] from the aspects of image prior and noise statistics. Due to limited space, we do not introduce many equally excellent image denoising methods [23–33].

Category	Example	Image Prior		Noise Statistics
		Internal	External	
Supervised Networks	Noise2Clean ([8, 9, 11–13, 22], etc.)	✓	✓	✓
Unsupervised Networks	Noise2Noise [18]	✓	✓	✓
	Deep Image Prior [19]	✓	✓	
	Noise2Void [20]	✓		
	Noise2Self [21]	✓		
	Noisy-As-Clean (Ours)	✓		✓

Table 1: **Summary of different networks for image denoising.** The networks are divided into two categories: supervised networks and unsupervised ones. In each category, some representative works are listed as examples. “Image Prior” includes internal and external priors. “Internal” image priors are directly learned on the input test image itself, without access to external images for training. “External” image priors need to be learned with external images (as long as not the test input). “Noise Statistics” indicates that the network is able to learn the noise statistics from the training data.

Supervised networks are trained with plenty pairs of noisy and clean images by minimizing a loss function. The process can be viewed as learning a mapping function from the noisy inputs to the clean targets. This category of networks can learn external image priors and noise statistics from the training data. Numerous deep neural network based methods [6–11, 13] have been developed with achieving promising performance on synthetic noise (e.g., AWGN). However, several supervised networks perform surprisingly worse than BM3D [24] when applied to noisy images in real scenarios [34–37].

Unsupervised networks for image denoising are developed mainly due to the lack of clean target images in real-world photographs captured by camera sensors. Along this direction, Noise2Noise [18] learns a mapping function between two images with the same scene, but independently sampled with different noise. This work is feasible to learn external image prior and noise statistics from the training data. However, its requirements are very challenging, since capturing two exactly the same signal in two independent samplings is really difficult in real-world scenarios. To alleviate this problem, Noise2Void [20] predicts a pixel from its surroundings by learning blind-spot networks,

so that the network only use noisy images during training. This work assumes that the corruption is zero-mean and independent between pixels. However, as Noise2Self [21] studied, Noise2Void significantly reduces the training efficiency, and reduces the denoising performance at test time. Recently, the work of Deep Image Prior [19] indicates that the structure of networks can resonate with the natural image priors, and can be utilized in image restoration without external training data. However, for each test corrupted image, this method needs to choose a suitable network and interrupt its training process at a right moment, which is usually unpredictable for different images.

Image priors are widely used for different image restoration tasks [25–27, 38–40]. They can be divided into internal priors and external priors. The internal image priors are directly learned on the input test image itself, while the external image priors need to be learned with external images (as long as not the test input). The internal priors are adaptive to its contents, but somewhat affected by the corruptions [25, 38, 39]. By contrast, the external priors are effective for restoring general images, but not optimal for specific test image [26, 27, 40].

Noise statistics is of the key importance for the image denoising task. The AWGN noise is one representative type with widespread study. Recently, more attention has been paid to the realistic noise generated in camera sensors. This type of noise is often modeled to be mixed Poisson and Gaussian distributed [15]. The Poisson component mainly comes from the irregular photons hitting the sensor [16], while the AWGN part is majorly produced by the dark current [14]. Though performing well on the synthetic noise being trained with, supervised denoisers [6–11, 13, 17] still suffer from the *domain gap dilemma* when handling the observed noisy image in real-world scenarios.

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

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

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

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

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 $p(\mathbf{y}_i, \mathbf{x}_i) = p(\mathbf{x}_i)p(\mathbf{y}_i|\mathbf{x}_i)$, we observe that training on whole sampled data can be decomposed into the same minimization problem at every pair of training images. Through simple manipulations, Eqn. (2) is equivalent to

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

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 [18, 20], here we study a practical assumption on noise statistics, i.e., *the expectation $\mathbb{E}[\mathbf{x}]$ and variance $\text{Val}[\mathbf{x}]$ of signal intensity are much stronger than those of noise $\mathbb{E}[\mathbf{n}_o]$ and $\text{Val}[\mathbf{n}_o]$* (such that they are negligible but not necessarily zero):

$$\mathbb{E}[\mathbf{x}] \gg \mathbb{E}[\mathbf{n}_o], \quad \text{Val}[\mathbf{x}] \gg \text{Val}[\mathbf{n}_o]. \quad (4)$$

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} :

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

Now we add *simulated noise* \mathbf{n}_s to the *observed* noisy image \mathbf{y} , and generate a new noisy image $\mathbf{z} = \mathbf{y} + \mathbf{n}_s$. Ideally, \mathbf{n}_s is with the same statistics to \mathbf{n}_o : $\mathbb{E}[\mathbf{n}_s] = \mathbb{E}[\mathbf{n}_o]$ and $\text{Val}[\mathbf{n}_s] = \text{Val}[\mathbf{n}_o]$. Then we have

$$\mathbb{E}[\mathbf{z}] \gg \mathbb{E}[\mathbf{n}_s], \quad \text{Val}[\mathbf{z}] \gg \text{Val}[\mathbf{n}_s]. \quad (6)$$

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

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

By the *Law of Total Expectation* (or *Tower Rule*) [41], we have

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

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^* :

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

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

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

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

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

The conclusion 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 noisy and clean image pairs $\{(\mathbf{y}_i, \mathbf{x}_i)\}$.*

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 (or simulated) noise \mathbf{n}_o (or \mathbf{n}_s) follows a mixed \mathbf{x} -dependent (or \mathbf{y} -dependent) Poisson distribution parameterized by λ_o (or λ_s) and Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma_o^2)$ (or $\mathcal{N}(\mathbf{0}, \sigma_s^2)$), i.e.,

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

where $\mathbf{x} \odot P(\lambda_o)$ 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

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

Here, ρ is the correlation between \mathbf{n}_o and \mathbf{n}_s , and $\rho = 0$ if they are independent from each other. 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 in the experimental section (§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

Based on the our statistical analysis, here we proposed to learn unsupervised networks with the “Noisy-As-Clean” 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 . The idea is illustrated in Figure 1.

Training stage. In real-world images captured by camera sensors, one can hardly separate the noise from the signal, observation is that the signal intensity \mathbf{x} is usually stronger than the noise intensity. That is, the expectation of the noise \mathbf{n}_o is usually much smaller than the latent clean image \mathbf{x} . We can observe that, if we train a 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 as well as the image-specific noise statistics. It has the power to remove the noise \mathbf{n}_s from the new noisy image \mathbf{z} . Then if we perform denoising on the original noisy

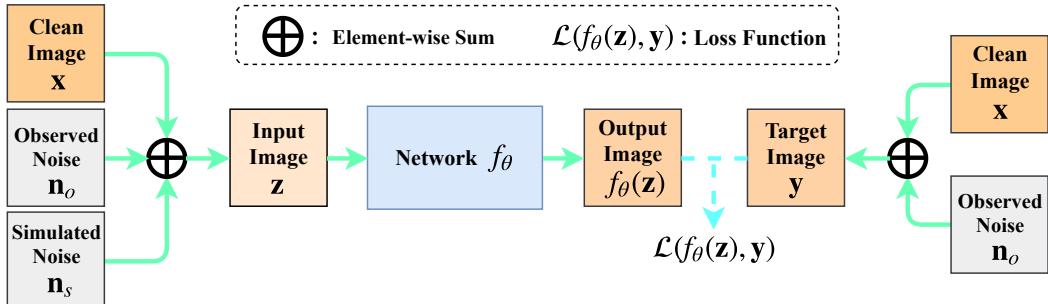


Figure 1: **Proposed “Noisy-As-Clean” strategy for training unsupervised image denoising networks.** In this strategy, we take the *observed* noisy image $y = x + n_o$ as the “clean” target, and take the *simulated* noisy image $z = y + n_s$ as the input. We do not regard the clean image x as target.

image y , then the realistic noise n_o can easily be removed. Note that we **do not** use any clean image x as target “ground-truth” in the training stage.

Testing is performed by directly regarding an *observed* noisy image $y = x + n_o$ as input. We only test the image y once. The denoised image can be represented as $\hat{y} = f_{\theta^*}(y)$, with which the objective metrics such as PSNR and SSIM [42] can be computed with the clean image x .

Implementation details. We employ the ResNet-50 network used in [19] as the backbone network (implemented in PyTorch [43]), which includes 10 residual blocks. Its parameters are randomly initialized *without being pretrained*. The optimizer is Adam [44] 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.

5 Experiments

In this section, we perform extensive experiments on image denoising to evaluate the denoising performance of the networks learned with the proposed “Noisy-As-Clean” (NAC) strategy. In all experiments, we train a denoising network using only the observed noisy image y as the target, and the *simulated* noisy image z (with data augmentation) as the input. For all comparison methods, the source codes or trained models are downloaded from the corresponding authors’ websites, and we use the default parameter settings, unless otherwise specified. The PSNR, SSIM [42] and visual quality of different methods are compared to evaluate the performance. We first test with simple noise statistics such as additive white Gaussian noise (AWGN), signal dependent Poisson noise, and mixed Poisson-AWGN noise in §5.1, and continue to more complex realistic noise in §5.2. In §5.3, we perform comprehensive ablation studies to gain deeper insights into the proposed NAC strategy.

5.1 Synthetic Noise Removal

We evaluate NAC networks on images corrupted by synthetic noise, including AWGN noise, signal dependent Poisson noise, and mixture of both. More results and comparisons on visual quality are provided in the *Supplementary File*.

Training NAC network is performed for each test image. Here, we train an image-specific denoising network using the observed noisy image y as the target, and the *simulated* noisy image z as the input. Each *observed* noisy image y is generated by adding the *observed* noise n_o to the clean image x . The *simulated* noisy image z is generated by adding *simulated* noise n_s to *observed* noisy image y .

Comparison methods. We compare our NAC networks with state-of-the-art image denoising methods [11, 18, 19, 24, 45]. On AWGN noise, we compare with BM3D [24], DnCNN [11], Noise2Noise (N2N) [18], and Deep Image Prior (DIP) [19]. On signal dependent Poisson noise, We compare with Inverse Anscombe Transform (ANSC) [45], N2N [18], and DIP [19]. On Poisson noise and mixed Poisson-AWGN noise, We compare with ANSC [45] and DIP [19]. For N2N [18], we retrain its networks on the *BSD300* dataset [46] for fair comparisons.

Testing datasets. We evaluate the comparison methods on the *Set12* and *BSD68* datasets, which are widely tested by supervised denoising networks [8, 9, 11, 13]. The *Set12* dataset contains 12 images of sizes 512×512 or 256×256 , while the *BSD68* dataset contains 68 images of different sizes. Most of the grayscale images are widely tested by traditional methods [24, 27, 31, 40].

Results on AWGN noise. We test AWGN noise with $\sigma \in \{5, 10, 15, 20, 25\}$, i.e., the observed noise \mathbf{n}_o is AWGN with standard deviation (std) of 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 [42] than BM3D [24] and DnCNN [11], two previous leading image denoising methods. Note that DnCNN are supervised networks trained on clean and synthetic noisy image pairs. When compared to the unsupervised methods such as N2N [18] and DIP [19], the networks trained by the proposed NAC strategy outperform by a larger margin on PSNR and SSIM.

Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$		$\sigma = 25$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑
BM3D [24]	38.07	0.9580	34.40	0.9234	32.38	0.8957	31.00	0.8717	29.97	0.8503
DnCNN [11]	38.76	0.9633	34.78	0.9270	32.86	0.9027	31.45	0.8799	30.43	0.8617
N2N [18]	40.82	0.9665	36.95	0.9446	33.99	0.9149	31.80	0.8788	30.72	0.8446
DIP [19]	32.49	0.9344	31.49	0.9299	29.59	0.8636	27.67	0.8531	25.82	0.7723
NAC (Ours)	41.43	0.9794	37.31	0.9530	34.67	0.9206	32.92	0.8849	31.40	0.8459

Table 2: Average PSNR (dB) and SSIM [42] results of different methods on *Set12* datasets corrupted by AWGN noise. The best results are highlighted in bold.

Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$		$\sigma = 25$	
	Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
BM3D [24]	37.59	0.9640	33.32	0.9163	31.07	0.8720	29.62	0.8342	28.57	0.8017
DnCNN [11]	38.07	0.9695	33.88	0.9270	31.73	0.8706	30.27	0.8563	29.23	0.8278
N2N [18]	38.58	0.9627	34.07	0.9200	31.81	0.8770	30.14	0.8550	28.67	0.8123
DIP [19]	29.74	0.8435	28.16	0.8310	27.07	0.7867	25.80	0.7205	24.63	0.6680
NAC (Ours)	39.00	0.9707	34.66	0.9293	32.07	0.8779	30.33	0.8275	28.89	0.7798

Table 3: Average PSNR (dB) and SSIM [42] results of different methods on *BSD68* corrupted by AWGN noise. The best results are highlighted in bold.

Results on Poisson noise. Here, the observed noise \mathbf{n}_o is signal dependent Poisson distributed with parameter $\lambda \in \{5, 10, 15, 20, 25\}$. During training, the simulated noise \mathbf{n}_s is with the same λ as that of \mathbf{n}_o . Note that \mathbf{n}_s is y -dependent, which is different with the x -dependent \mathbf{n}_o . The comparison results are listed in Tables 4 and 5. It can be seen that, the network trained with the proposed NAC strategy achieve much better performance on PSNR and SSIM [42] than ANSC [45], the state-of-the-art method for signal dependent Poisson noise removal. When compared to the unsupervised methods such as N2N [18] and DIP [19], the proposed NAC networks achieve prior results by a large margin on average PSNR and SSIM. This demonstrates the advantages of the proposed NAC strategy on removing Poisson noise.

Noise Level	$\lambda = 5$		$\lambda = 10$		$\lambda = 15$		$\lambda = 20$		$\lambda = 25$	
	Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
ANSC [45]	40.26	0.9478	38.03	0.9311	35.68	0.9196	34.19	0.9082	32.85	0.8974
N2N [18]	41.27	0.9679	38.97	0.9483	36.68	0.9303	34.49	0.9205	32.78	0.9116
DIP [19]	31.31	0.8725	29.92	0.8458	27.95	0.8145	26.89	0.7860	25.40	0.7554
NAC (Ours)	44.20	0.9923	41.01	0.9813	38.75	0.9701	37.10	0.9579	35.88	0.9445

Table 4: Average PSNR (dB) and SSIM [42] results of different methods on *Set12* dataset corrupted by signal dependent Poisson noise. The best results are highlighted in bold.

Results on mixed Poisson-AWGN noise. Each noisy image is generated by adding mixed Poisson-AWGN noise. Here, the parameters λ and σ for Poisson and AWGN noise, respectively, are set as $\{(\lambda = 5, \sigma = 15), (\lambda = 5, \sigma = 25), (\lambda = 15, \sigma = 5), (\lambda = 25, \sigma = 5)\}$. From Tables 6 and 7, one can see that the network trained with the proposed NAC strategy achieve much better performance on PSNR and SSIM [42] than ANSC [45], the current state-of-the-art method for signal dependent Poisson noise removal. When compared to the unsupervised method DIP [19], one can see that the networks trained by the proposed NAC strategy outperform by a larger margin on PSNR and SSIM. Since N2N [18] did not deal with this type of noise, here we do not compare with this method.

Noise Level	$\lambda = 5$		$\lambda = 10$		$\lambda = 15$		$\lambda = 20$		$\lambda = 25$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑
ANSC [45]	35.06	0.9206	31.89	0.8913	29.74	0.8591	27.52	0.8078	26.43	0.7455
N2N [18]	36.56	0.9526	32.04	0.9162	30.56	0.8648	28.64	0.8142	27.19	0.7681
DIP [19]	23.97	0.7814	22.79	0.7283	21.78	0.6796	21.22	0.6570	20.77	0.6360
NAC (Ours)	38.89	0.9671	34.57	0.9296	31.96	0.8842	29.98	0.8282	28.47	0.7757

Table 5: Average PSNR (dB)/SSIM results of different methods on *BSD68* dataset corrupted by signal dependent Poisson noise. The best results are highlighted in bold.

Noise Level	$\lambda = 5, \sigma = 15$		$\lambda = 5, \sigma = 25$		$\lambda = 15, \sigma = 5$		$\lambda = 25, \sigma = 5$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑
ANSC [45]	31.38	0.8521	29.52	0.8117	35.21	0.9303	33.54	0.9029
DIP [19]	23.07	0.7686	22.59	0.7449	24.25	0.7901	21.74	0.7060
NAC (Ours)	34.48	0.9122	31.20	0.8370	37.67	0.9632	35.21	0.9424

Table 6: Average PSNR (dB) and SSIM [42] results of different methods on *Set12* dataset corrupted by mixed Poisson-AWGN noise. The best results are highlighted in bold.

5.2 Practice on Real Photographs

With the promising performance on synthetic noise removal, here we tackle the realistic noise for practical applications. The *observed* realistic noise n_o can be roughly modeled as mixed Poisson noise and AWGN noise [15, 17]. Hence, for each *observed* noisy image y , we generate the *simulated* noise n_s by sampling the y -dependent Poisson part and the independent AWGN noise.

Training NAC network 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 n_o . Therefore, the *simulated* noise 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 parameter λ in $0 < \lambda \leq 25$, and the AWGN noise is randomly sampled with parameter σ in $0 < \sigma \leq 25$.

Comparison methods. We compare with state-of-the-art methods on real-world image denoising, including CBM3D [47], the commercial software Neat Image [48], two supervised networks DnCNN+ [49] and CBDNet [17], and two unsupervised networks Noise2Noise [18], DIP [19]. Note that DnCNN+ [49] and CBDNet [17] are two state-of-the-art supervised networks for real-world image denoising, and DnCNN+ is an improved extension of DnCNN [11] (the authors provide us the models/results of DnCNN+).

Testing dataset. We evaluate the comparison methods on the Cross-Channel dataset [14], which includes noisy images of 11 static scenes captured by Canon 5D Mark 3, Nikon D600, and Nikon D800 cameras. The real-world noisy images were 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.

Results on PSNR and SSIM. The comparisons on average PSNR and SSIM results are listed in Table 8. As can be seen, the proposed NAC networks achieve better performance than all previous denoising methods, including the CBM3D [47], the supervised networks DnCNN+ [11] and CBDNet [17], and the unsupervised networks N2N and DIP [19]. 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+ [11] and CBDNet [17].

Noise Level	$\lambda = 5, \sigma = 15$		$\lambda = 5, \sigma = 25$		$\lambda = 15, \sigma = 5$		$\lambda = 25, \sigma = 5$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑
ANSC [45]	29.97	0.8343	27.68	0.7874	29.34	0.8618	26.15	0.7629
DIP [19]	23.79	0.7314	21.99	0.6264	22.54	0.6900	21.03	0.6291
NAC (Ours)	31.36	0.8542	28.92	0.8080	31.67	0.8802	28.28	0.7736

Table 7: Average PSNR (dB) and SSIM results of different methods on *BSD68* dataset corrupted by mixed Poisson-AWGN noise. The best results are highlighted in bold.

Speed. The work most similar to ours is Deep Image Prior (DIP) [19], which also trains an image-specific network for each test image. Averagely, DIP needs 909.2 seconds to process a 512×512 color image, on which our NAC network only needs 982.4 seconds (on an NVIDIA Titan X GPU).

Type	Traditional Methods		Supervised Networks		Unsupervised Networks		
Method	CBM3D [47]	NI [48]	DnCNN+ [11]	CBDNet [17]	N2N [18]	DIP [19]	NAC
PSNR↑	35.19	35.33	35.40	36.44	35.32	35.69	36.59
SSIM↑	0.9063	0.9212	0.9115	0.9460	0.9160	0.9259	0.9502

Table 8: Average PSNR (dB) and SSIM [42] results of different methods on the *default* 15 cropped real-world noisy images in Cross-Channel dataset [14]. The best results are highlighted in bold.

5.3 Ablation Study

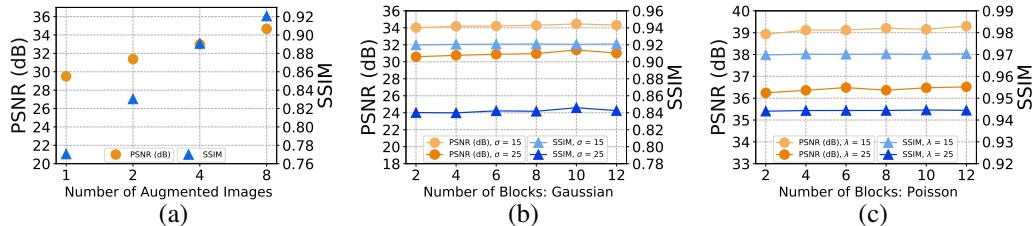


Figure 2: PSNR (dB) and SSIM results on the *Set12* dataset by NAC networks w.r.t.: (a) number of augmented images on AWGN noise removal ($\sigma = 15$); (b) number of residual blocks on AWGN noise removal ($\sigma = 15, 25$); and (c) number of residual blocks on Poisson noise removal ($\lambda = 15, 25$).

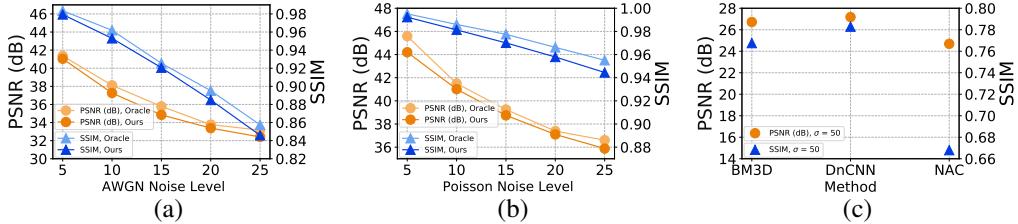


Figure 3: Comparisons with the Oracle version of NAC networks trained on the pair of *observed* noisy image y and the clean image x in the *Set12* dataset when (a) corrupted by AWGN noise with $\sigma = 5, 10, 15, 20, 25$; (b) corrupted by Poisson noise with $\lambda = 5, 10, 15, 20, 25$. (c) PSNR (dB) and SSIM results by BM3D [24], DnCNN [11], and the proposed NAC networks performed on the *Set12* dataset corrupted by strong AWGN noise ($\sigma = 50$).

Data augmentation is essential since our NAC strategy is statistically effective, requiring multiple simulated noisy images $\{z\}$ for each *observed* noisy image y . Here, we study the performance of NAC on image denoising trained with 1 input (the original z), 2 inputs (z with its horizontal reflection), 4 inputs (z in 4 different rotations), and 8 inputs (all 8 transformations). From Figure 2 (a), one can see that with more transformations, the NAC networks achieve better performance on PSNR and SSIM over the *Set12* dataset for AWGN noise removal (here we use $\sigma = 15, 25$).

Number of residual blocks. Our backbone network is the ResNet-50 [19] including 10 residual blocks, each block is of 5-layer consisting of ‘Conv-BN-ReLU-Conv-BN’. Now we study how the number of blocks influence the performance of NAC on image denoising. From Figure 2 (b) and Figure 2 (c), one can see that, with different number of residual blocks, the NAC networks achieve similar PSNR and SSIM performance on the *Set12* dataset corrupted by AWGN noise ($\sigma = 15, 25$) and signal dependent Poisson noise ($\lambda = 15, 25$), respectively. This indicates that the number of blocks has little influence on PSRN and SSIM metrics. Hence, we use 10 blocks the same as [19].

Comparison with Oracle. We also study the ‘Oracle’ performance of our NAC networks. The experiments are performed on *Set12* dataset corrupted by AWGN or signal dependent Poisson noise. The noise levels are in $\{5, 10, 15, 20, 25\}$. In ‘Oracle’, we trained our NAC networks on the pair of *observed* noisy image y and its clean image x corrupted by AWGN noise or signal dependent Poisson noise. Figure 3 (a) and Figure 3 (b) show comparisons with the ‘Oracle’ networks on PSNR

and SSIM. It can be seen that, the “Oracle” networks trained on noisy-clean image pairs (\mathbf{y}, \mathbf{x}) only perform slightly better than the original NAC networks trained with the *simulated-observed* noisy image pairs (\mathbf{z}, \mathbf{y}). With the proposed NAC strategy, we can achieve similar denoising performance only with the *observed* noisy image \mathbf{y} , at small noise levels.

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 [24] and DnCNN [11], on *Set12* corrupted by AWGN noise with $\sigma = 50$. The PSNR and SSIM results are plotted in Figure 3 (c). One can see that, our NAC networks are limited in handling strong AWGN noise, when compared to the BM3D [24] and DnCNN [11].

6 Conclusion

In this work, we proposed a “Noisy-As-Clean” (NAC) strategy for learning unsupervised image denoising networks. In NAC, we take an *observed* noisy image as target, and generate *simulated* noisy images by adding simulated noise to it. The simulated noise is close to the noise in *observed* noisy image. This strategy can be seamlessly embedded into existing supervised denoising networks. We provided a statistically useful conclusion: it is possible to learn an unsupervised network only with an *observed* noisy image, approximating the optimal parameters of a supervised network learned with pairs of noisy and clean images. Comprehensive experiments on benchmark datasets demonstrate that, the networks trained with the proposed NAC strategy can achieve better performance than previous supervised learning based networks on image denoising. The image-specific NAC networks can better capture the specific image priors and noise statistics from the *observed* noisy image to be restored.

7 Appendix: Comparisons on the visual quality of denoised images by different methods

Here, we compare the visual quality of the denoised images by different methods on synthetic AWGN noise and realistic noise on the *Set12*, *BSD68*, and *Cross-Channel* [14] datasets. The synthetic noise removal are evaluated on *Set12* and *68* datasets, while the realistic noise removal are performed on the *Cross-Channel* dataset. For each comparison, the PSNR and SSIM [42] results are provided for reference. We compare with BM3d [24], DnCNN [11], N2N [18], and DIP [19] on synthetic AWGN noise removal; ANSC [45], N2N [18], and DIP [19] on synthetic Poisson noise removal; ANSC [45] and DIP [19] on mixed Poisson-AWGN noise removal; CBM3D [47], NI [48], DnCNN+ [11], CBDNet [17], N2N [18], and DIP [19] on realistic noise removal. From Figures 4–7, one can see that, in all cases, the proposed NAC networks preserve the image details (signals) better than other methods, while other methods remove out these small but visually important signals.

References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
- [2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.
- [6] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR*, pages 2392–2399, 2012.
- [7] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, pages 341–349, 2012.
- [8] X. Mao, C. Shen, and Y. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. In *NIPS*, 2016.

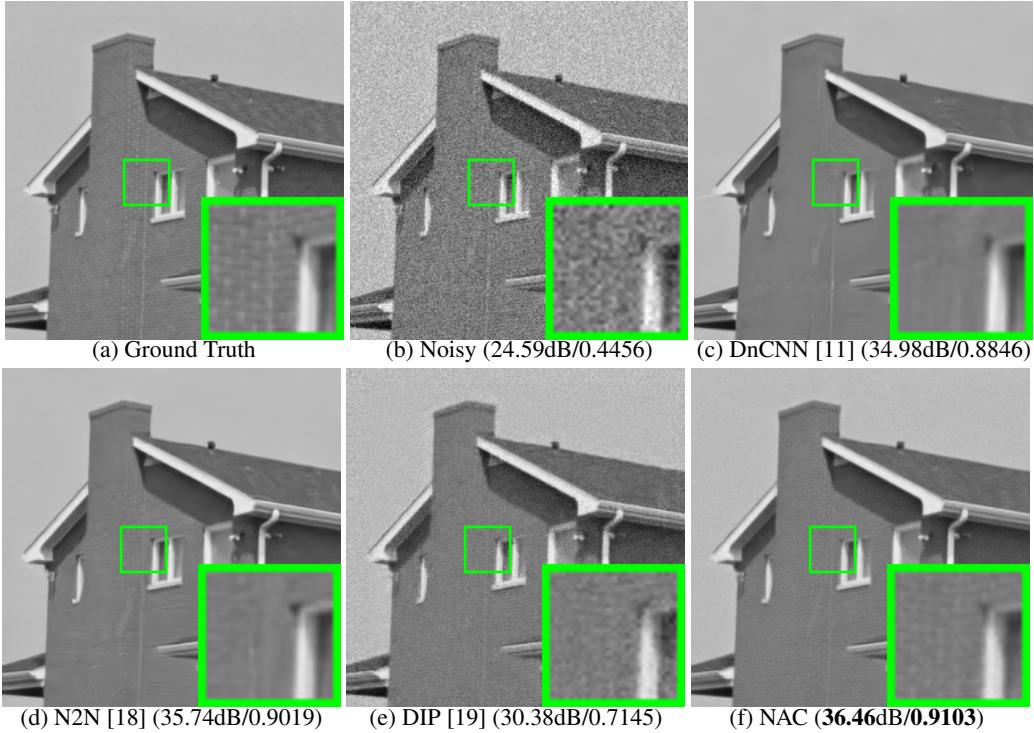


Figure 4: **Denoised images and corresponding PSNR/SSIM results of “House” in Set12 by different methods.** The images are corrupted by AWGN noise with $\sigma = 15$. The best results on PSNR and SSIM are highlighted in bold.

- [9] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017.
- [10] S. Lefkimiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, pages 3587–3596, 2017.
- [11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 2017.
- [12] T. Plötz and S. Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018.
- [13] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang. Non-local recurrent network for image restoration. In *NeurIPS*, pages 1673–1682. 2018.
- [14] 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 *CVPR*, pages 1683–1691, 2016.
- [15] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, Oct 2008.
- [16] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. *CVPR*, 1:901–908, 2006.
- [17] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019.
- [18] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, pages 2971–2980, 2018.
- [19] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018.
- [20] A. Krull, T. Buchholz, and F. Jug. Noise2Void-learning denoising from single noisy images. *arXiv preprint arXiv:1811.10980*, 2018.
- [21] J. Batson and L. Royer. Noise2Self: Blind denoising by self-supervision. *arXiv preprint arXiv:1901.11365*, 2019.
- [22] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. In *CVPR*, pages 9446–9454, 2019.
- [23] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *CVPR*, pages 60–65, 2005.

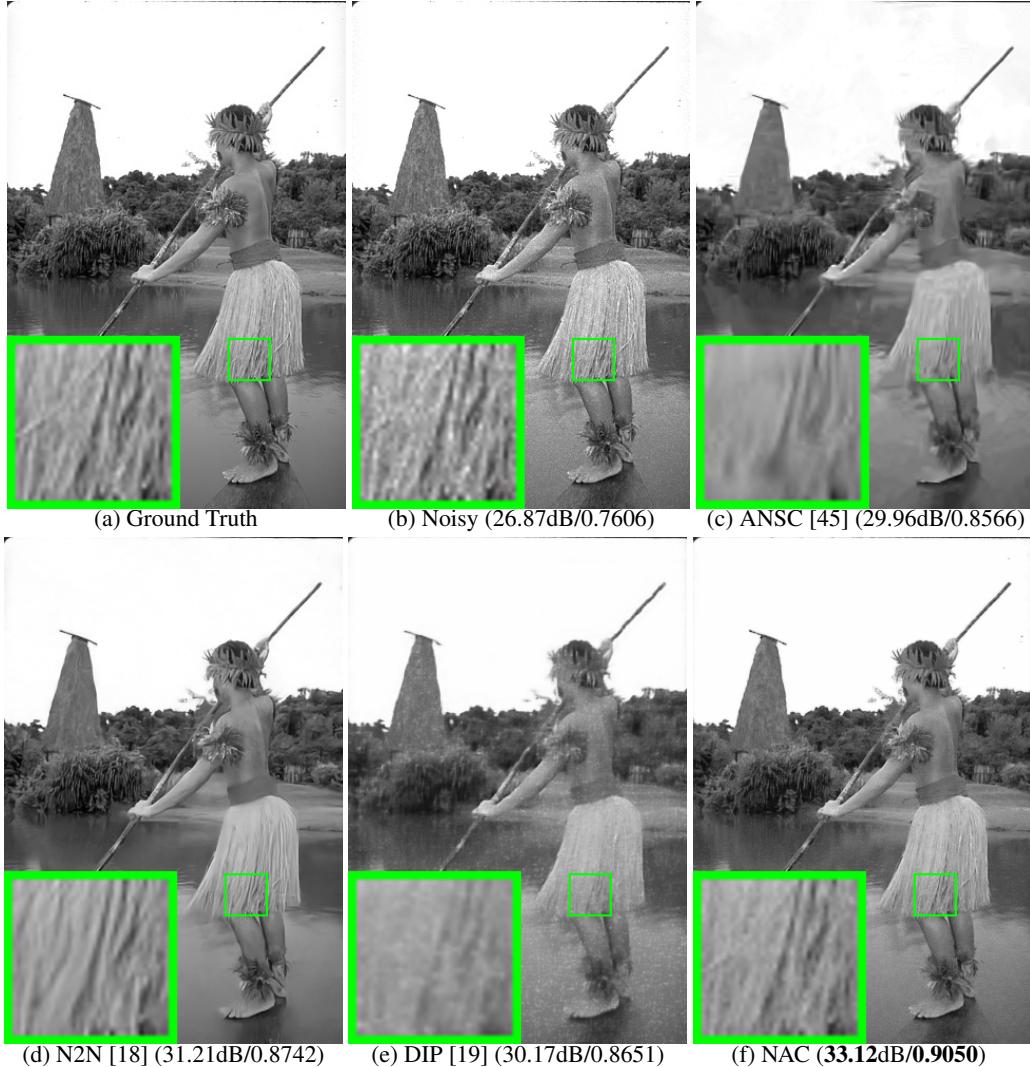


Figure 5: **Denoised images and corresponding PSNR/SSIM results of “test002” in **BSD68** by different methods.** The images are corrupted by signal dependent Poisson noise with $\lambda = 15$. The best results on PSNR and SSIM are highlighted in bold.

- [24] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.
- [25] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.
- [26] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.
- [27] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011.
- [28] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.
- [29] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, June 2014.
- [30] Y. Chen and T. 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.
- [31] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, 121(2):183–208, 2017.

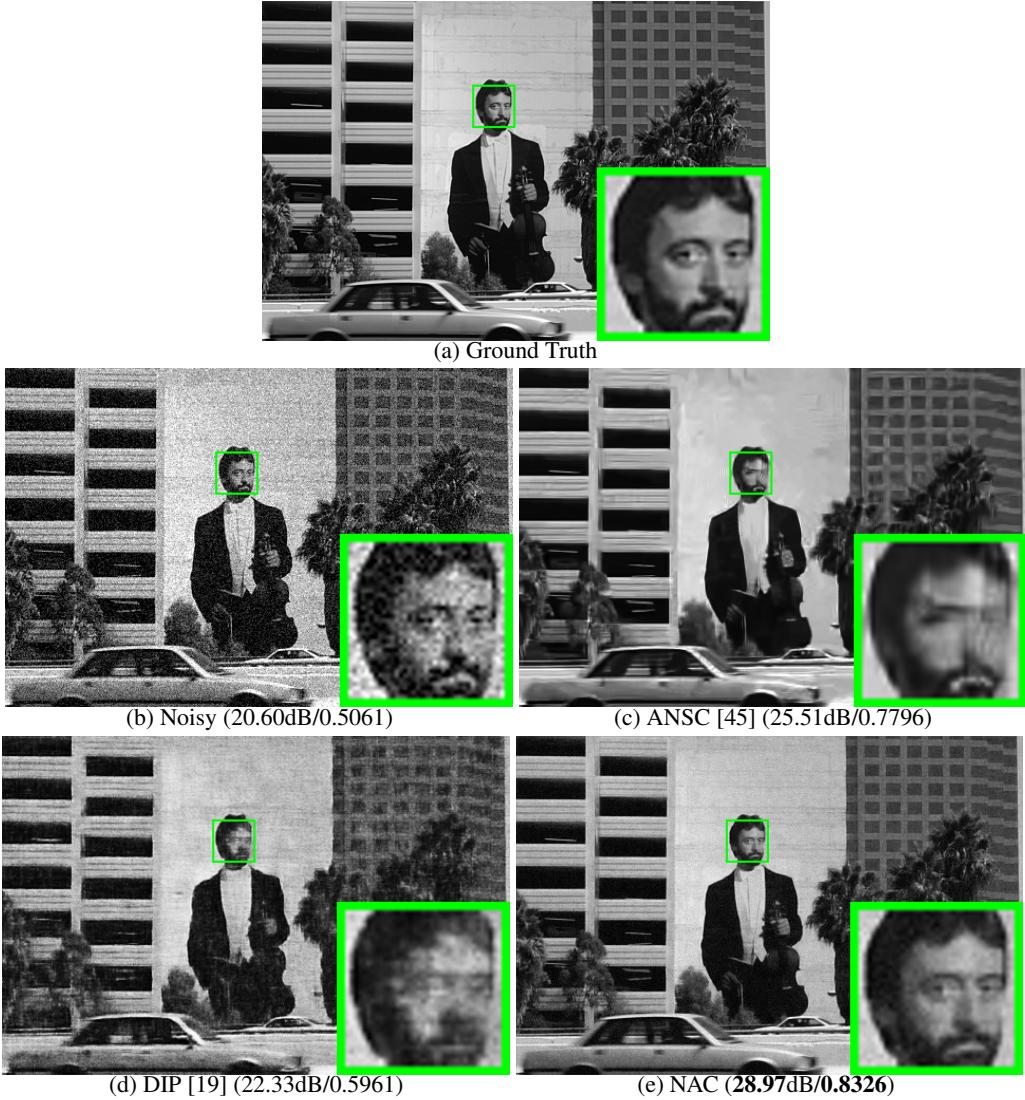


Figure 6: **Denoised images and corresponding PSNR/SSIM results of “test011” in Set12 by different methods.** The images are corrupted by mixed Poisson and AWGN noise with $\lambda = 5, \sigma = 25$. The best results on PSNR and SSIM are highlighted in bold.

- [32] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, 2017.
- [33] J. Xu, L. Zhang, and D. Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *ECCV*, 2018.
- [34] Darmstadt Noise Dataset Benchmark. https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb. Accessed: 2019-05-23.
- [35] Smartphone Image Denoising Dataset Benchmark. <https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php>. Accessed: 2019-05-23.
- [36] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.
- [37] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, June 2018.
- [38] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *NIPS*, pages 2295–2303. 2009.
- [39] M. Zontak and M. Irani. Internal statistics of a single natural image. In *CVPR*, 2011.
- [40] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *ICCV*, pages 244–252, 2015.

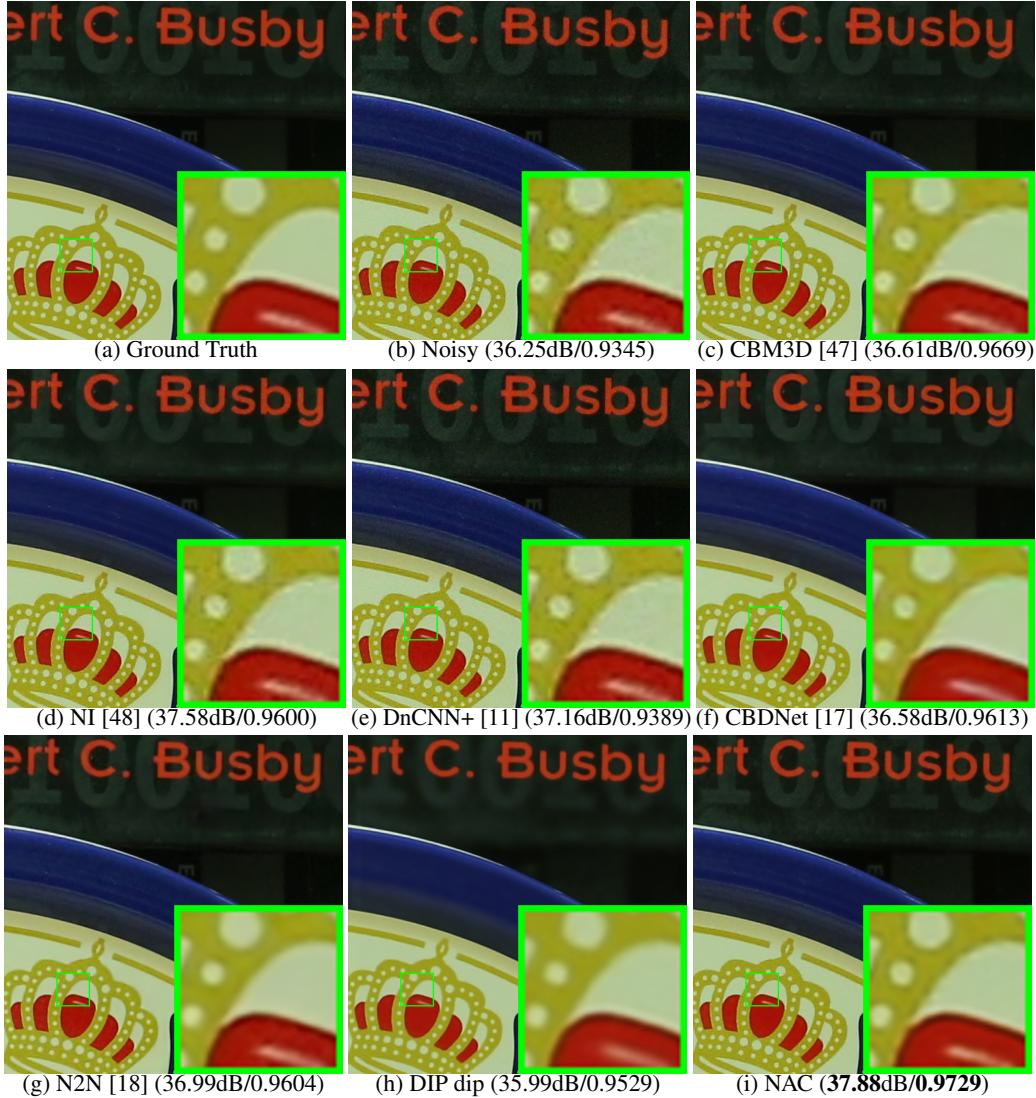


Figure 7: Denoised images and corresponding PSNR/SSIM results of “5dmark3-iso3200-1” in the **Cross-Channel** dataset [14] by different methods. The best results are highlighted in bold.

- [41] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995.
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [43] PyTorch. <https://pytorch.org/>. Accessed: 2019-05-23.
- [44] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [45] M. Makitalo and A. Foi. Optimal inversion of the anscombe transformation in low-count poisson image denoising. *IEEE Transactions on Image Processing*, 20(1):99–109, 2011.
- [46] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.
- [47] 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 *ICIP*, pages 313–316. IEEE, 2007.
- [48] Neatlab ABSoft. Neat Image. <https://ni.neatvideo.com/home>.
- [49] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 2018.