Recorrupted-to-Recorrupted: Unsupervised Deep Learning for Image Denoising

Tongyao Pang[1], Huan Zheng[1], Yuhui Quan[2], and Hui Ji[1]
[1]Department of Mathematics, National University of Singapore, 119076, Singapore
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
matpt@nus.edu.sg, huan_zheng@u.nus.edu,csyhquan@scut.edu.cn, and matjh@nus.edu.sg

Abstract

Deep denoiser, the deep network for denoising, has been the focus of the recent development on image denoising. In the last few years, there is an increasing interest on developing unsupervised deep denoisers which only call unorganized noisy images without ground truth for training. Nevertheless, the performance of these unsupervised deep denoisers is not competitive to their supervised counterparts. Aiming at developing a more powerful unsupervised deep denoiser, this paper proposed a data augmentation technique, called recorrupted-to-recorrupted (R2R), to address the overfitting caused by the absence of truth images. For each noisy image, we showed that the cost function defined on the noisy/noisy image pairs constructed by the R2R method is statistically equivalent to its supervised counterpart defined on the noisy/truth image pairs. Extensive experiments showed that the proposed R2R method noticeably outperformed existing unsupervised deep denoisers, and is competitive to representative supervised deep denoisers.

1. Introduction

Image denoising is one fundamental problem in image processing which receives an enduring interest in last decades. It aims at removing random noise from the input images to improve their signal-to-noise-ratios (SNRs). Image denoising is not only an important problem itself but also serves as a basic module in many image recovery methods. A noisy image is usually formulated as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{y}$ denotes the noisy image, $\boldsymbol{x}$ the noise-free image for recovery, and $\boldsymbol{n}$ measurement noise. The noise $\boldsymbol{n}$ is often assumed to be the instance drawn from some distribution.

In recent years, deep learning is the main driving force in the development of image denoisers. A majority of existing deep-learning-based denoising methods (*e.g.* [31, 36, 37]) are supervised, which learn the mapping from noisy input to its clean counterpart by training a deep neural network (DNN) on many clean/noisy image pairs. However, in order to have a trained model that generalizes well, a great amount of such noisy/clean image pairs are needed to sufficiently cover the variations on image content and measurement noise. Fulfilling such a demanding requirement on training samples may be costly and sometimes challenging. For example, it is non-trivial to collect real-world noisy/clean image pairs; see *e.g.* [25, 33, 3]. For scientific images and medical images, the task is more challenging.

Recently, it is receiving an increasing interest on relaxing the prerequisite of supervised learning on training samples. Lehtinen *et al*. [21] presented a weakly supervised method, the so-called Noise2Noise method, which directly trains the DNN between the pairs of two noisy images of the same scene. As the noise of such image pairs is independent, the expectation of the cost function of Noise2Noise is then the same as that of the supervised one defined on the noisy/truth image pairs. However, collecting noisy image pairs of the same scene remains highly non-trivial as image alignment can be an issue, and it is not possible for the images of dynamic scenes. More recent works on unsupervised deep denoisers have been focusing on training DNNs using noisy image dataset without pair-wise correspondence, or even training DNNs only using only the input noisy image itself. These methods can be categorized to two classes.

- *Data augmentation methods*. Noise2Void [17] and Noise2Self [5] adopt the blind-spot strategy to avoid overfitting (convergence to identity map) when training a DNN to map a noisy image to itself, while Noiser2Noise [23] and Noise-as-Clean [32] add additional noise to the original noisy image to make image pairs which are then used to train the DNN.
- *Regularized denoising DNN*. The Stein's Unbiased Risk Estimator (SURE) [29, 22] regularizes the DNN by penalizing the divergence of the prediction. Deep image prior [30] uses early-stopping to avoid the overfitting. In Self2Self [26], a dropout-based training/testing scheme is introduced to reduce the bias and variance of the prediction from the DNN trained on single noisy image.

1.1. Motivation

Despite the great progress in last few years, the performance of unsupervised learning methods for denoising is still not comparable to that of their supervised counterparts, *e.g*. DnCNN [36] trained on noisy/clean pairs or Noise2Noise trained on noisy/noisy pairs. Indeed, many of them cannot compete well against classical non-local denoising method such as BM3D [11]. So far, SURE [29] provided the state-of-the-art (SOTA) performance among dataset-based unsupervised denoiser, and Self2Self [26] provided the SOTA performance among single-image-based unsupervised denoiser. In summary,

- Unsupervised learning has its value in many real-world applications, since it remains useful when no ground-truth image is available.
- Most existing unsupervised learning methods have a noticeable performance gap to their supervised counterparts, especially for denoising real-world images.

This paper aims at developing an unsupervised learning method for denoising that works on a set of unorganized noisy images without truth images. The proposed method not only provides the SOTA performance among existing unsupervised learning methods, but also is very competitive to many supervised learning methods including DnCNN.

1.2. Main idea

Revisiting Noise2Noise. Before proceeding, we take a revisit to Noise2Noise, the first attempt that relaxes the requirement of supervised denoising methods on training dataset: from noisy/clean image pairs to noisy/noisy image pairs. It is shown in [21] that the performance of a denoising network trained on noisy/noisy image pairs is roughly the same as that trained on noisy/clean image pairs of the same scene. Mathematically speaking, in the setting of additive Gaussian white noise (AGWN), a pair of noisy images of the same scene can be expressed as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} + \boldsymbol{n}, \qquad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma_1^2 \boldsymbol{I}), \\ \boldsymbol{y}' &= \boldsymbol{x} + \boldsymbol{n}', \quad \boldsymbol{n}' \sim \mathcal{N}(\boldsymbol{0}, \sigma_2^2 \boldsymbol{I}). \end{aligned}$$

Let $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$ denote the denoising DNN. Then, Noise2Noise trains the DNN by minimizing the squared-ℓ_2 loss:

$$\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{y}'\|_2^2\}. \tag{2}$$

Such a loss function is closely related to the one used in supervised learning:

$$\mathbb{E}_{\boldsymbol{n}}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{x}\|_2^2\}. \tag{3}$$

Indeed, we have

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{y}'\|_2^2\} \\ =&\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{x} - \boldsymbol{n}'\|_2^2\} \\ =&\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 - 2(\boldsymbol{n}')^\top(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{x}) + (\boldsymbol{n}')^\top\boldsymbol{n}'\} \\ =&\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{x}\|_2^2\} - 2\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{(\boldsymbol{n}')^\top\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y})\} + \text{const}. \end{aligned}$$

As long as the noise $\boldsymbol{n}$ and $\boldsymbol{n}'$ are independent, which gives $\mathbb{E}_{\boldsymbol{n},\boldsymbol{n}'}\{(\boldsymbol{n}')^\top\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y})\} = 0$, the expectation of the loss function defined on $(\boldsymbol{y}, \boldsymbol{y}')$ will be equivalent to the supervised one defined on $(\boldsymbol{y}, \boldsymbol{x})$ up to a constant. This is the reason why Noise2Noise can perform comparably to its supervised counterparts.

Re-corrupting both the input and target image for training on unorganized noisy images. Different from the dataset required by Noise2Noise, we only assume the availability of a set of unorganized noisy images without pairwise correspondence. In order to achieve comparable performance to Noise2Noise, the question is then about how to construct a pair of noisy images $(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})$ with independent noise from a single noisy image $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ such that

$$\mathbb{E}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}) - \widetilde{\boldsymbol{y}}\|_2^2\} = \mathbb{E}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}) - \boldsymbol{x}\|_2^2\} + \text{const}.$$

In the setting of AWGN: $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, our answer to the above question is to recorrupt the noisy image $\boldsymbol{y}$ as follows:

$$\widehat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{D}^\top\boldsymbol{z}, \quad \widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{D}^{-1}\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}), \tag{4}$$

where $\boldsymbol{D}$ can be any invertible matrix. We showed in Corollary 2 (Section 3) that the noise in $\widehat{\boldsymbol{y}}$ and $\widetilde{\boldsymbol{y}}$ are independent from each other, and thus the squared-ℓ_2 loss function trained on the image pair $(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})$ satisfies

$$\mathbb{E}_{\boldsymbol{n},\boldsymbol{z}}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}) - \widetilde{\boldsymbol{y}}\|_2^2\} = \mathbb{E}_{\widehat{\boldsymbol{n}}}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) - \boldsymbol{x}\|_2^2\} + \text{const}, \tag{5}$$

where $\widehat{\boldsymbol{n}} = \boldsymbol{n} + \boldsymbol{D}^\top\boldsymbol{z}$. Consider a dataset of un-organized noisy images

$$\boldsymbol{y}^k = \boldsymbol{x}^k + \boldsymbol{n}^k, \ \boldsymbol{x}^k \sim \mathcal{X}, \boldsymbol{n}^k \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}), k \in \mathbb{N}.$$

The cost function defined on the pairs $\{(\widehat{\boldsymbol{y}}^k, \widetilde{\boldsymbol{y}}^k)\}_{k\in\mathbb{N}}$ constructed by (4) is then equivalent to the following cost function:

$$\mathbb{E}_{\boldsymbol{x},\widehat{\boldsymbol{n}}}\{\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) - \boldsymbol{x}\|_2^2\} + \text{const},$$

i.e., the one used in the supervised learning on a set of noisy/truth image pairs $\{(\boldsymbol{x}^k + \widehat{\boldsymbol{n}}, \boldsymbol{x}^k)\}_{k\in\mathbb{N}}$.

Discussion. From (5), it can be seen that the proposed scheme (4) of the image pair $(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})$ leads to a loss function in the same form as that of Noise2Noise. Therefore, the network trained using the proposed scheme can be expected to have comparable performance to those supervised learning methods. Through this paper, the training scheme (5)

built on the construction scheme of image pair (4) is called *Recorrupted-to-Recorrupted*, abbreviated as R2R.

Moreover, the proposed R2R scheme also works for the noise which is signal-dependent. Suppose the noise follows a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{x}})$ with the x-dependent covariance matrix $\mathbf{\Sigma}_{\boldsymbol{x}}$. Then, one only needs to modify the recorruption scheme as follows:

$$\widehat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{D}^{\top}\boldsymbol{z}, \quad \widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{D}^{-1}\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{x}}).$$

The modified scheme above still leads to the same result as (5); See Section 3 for more details.

1.3. Contributions

In this paper, we proposed an unsupervised deep learning method for image denoising, named as R2R, which is trained on a dataset of un-organized noisy images, without truth or pair-wise correspondence. The contributions are summarized as follows:

- With rigorous mathematical treatment, this paper presented a so-called R2R unsupervised learning technique for image denoising, which is statistically equivalent to the supervised learning on noisy/clean image pairs.
- In comparison to other unsupervised learning methods for denoising, the proposed R2R is simple and flexible. It can be trained on external training samples or directly trained on noisy images for processing.
- Extensive experiments on synthetic noisy images show that the proposed R2R method performs better than all compared non-learning and unsupervised learning methods, and is comparable to representative supervised denoisers. For denoising real-world images, it is also very competitive to the top performers among the non-learning and unsupervised methods.

2. Related Work

There is abundant literature on image denoising, and we only focus on the most related ones.

Non-learning image denoisers based on image priors. In the past, many image denoisers have been proposed by imposing certain pre-defined image priors on clean images. Some widely-used image priors include: 1) Sparsity of image gradients, which leads to ℓ_p-norm penalization methods, *e.g.* [8, 28]; 2) Similarity of image patches, which induces non-local methods, *e.g.* BM3D [11] and WNMM [12], or rank-based regularization such as [14].

Supervised learning on noisy/truth image pairs. Supervised deep learning has been a prominent tool for image denoising, which trains denoising DNNs on many noisy/truth image pairs, *e.g.* [36, 37, 20, 13, 15, 4, 35, 27]. The DNNs are trained to map noisy images to their clean counterparts. DnCNN [36], which uses a residual convolutional DNN for training, is one widely-used method for benchmarking deep image denoisers. Instead of using noisy/truth image pairs, the Noise2Noise (N2N) method [21] is weakly supervised that trains the DNN on pairs of independent noisy images of the same scene.

Unsupervised learning on unpaired noisy images. Without noisy/truth image pairs, one approach is to use generative adversarial network (GAN) to generate these pairs from unpaired data for training, *e.g.* [10, 7]. Another type of method directly trains the DNN on noisy data, and the focus is on how to avoid overfitting which sees the DNN convergence to the identity map. A SURE-based method [29] regularizes the DNN by penalizing the divergence of the prediction. Some other methods propose data augmentation schemes to avoid overfitting and our R2R method falls into this category. In the following, we will review most related data augmentation methods.

Noise2Void (N2V) [17] and Noise2Self (N2S) [5] are based on the blind-spot strategy that randomly drops some pixels of the input and predicts them using their remaining neighbours. Laine *et al.* [18] proposes a specific blind-spot architecture that excludes the center pixel in its receptive field. The blind-spot technique can be conceptually interpreted as recorrupting the noisy sample by multiplicative Bernoulli noise. The issue is that a lot of information is discarded when discarding image pixels. In contrast, the proposed R2R keeps all image pixels. It is equivalent to train the DNN in a supervised manner with a only slightly higher noise level (*e.g.* $D = \frac{1}{2}\boldsymbol{I}$ in (4)). As a result, the R2R can be trained with better performance.

Given the noisy image $\boldsymbol{y}$, Noisier2Noise and Noisy-as-Clean use a noisier image as input, which is synthesized by recorrupting $\boldsymbol{y}$ with the noise $\boldsymbol{z}$, and then the DNN is trained over the pair $(\boldsymbol{y} + \alpha\boldsymbol{z}, \boldsymbol{y})$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \alpha\boldsymbol{z}) - \boldsymbol{y}\|_2^2. \tag{6}$$

The connection between the loss function defined above to the supervised one is not clear. In comparison, taking $\boldsymbol{D} = \alpha\boldsymbol{I}$ in (4), the R2R trains the DNN on $(\boldsymbol{y} + \alpha\boldsymbol{z}, \boldsymbol{y} - \boldsymbol{z}/\alpha)$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \alpha\boldsymbol{z}) - (\boldsymbol{y} - \boldsymbol{z}/\alpha)\|_2^2, \tag{7}$$

which rigorously showed its statistical connection to supervised learning. Indeed, the denoiser obtained by minimizing (6) is $\mathbb{E}(\boldsymbol{y}|\widehat{\boldsymbol{y}})$ ($\widehat{\boldsymbol{y}} = \boldsymbol{y} + \alpha\boldsymbol{z}$). To reduce noise effect further, Noisier2Noise runs a post-process for correction:

$$\alpha^{-2}\big((1+\alpha^2)\mathbb{E}(\boldsymbol{y}|\widehat{\boldsymbol{y}}) - \boldsymbol{z}\big).$$

In contrast, our R2R obtains the ideal denoiser $\mathbb{E}(\boldsymbol{x}|\widehat{\boldsymbol{y}})$ directly owing to the equivalence to the supervised learning.

Partially-linear denoiser [16] considered training a denoiser over the image pairs similar to our R2R method, and

showed its connection to supervised linear denoisers. As a denoising DNN is typically non-linear, they proposed to penalize the non-linear structure of the DNN to approximate its supervised counterpart well. A two-stage training procedure is then developed to learn such a denoiser with special structure. In comparison, the proposed cost function in our R2R method can use standard optimization procedure to train the network.

Self-supervised learning on single noisy image. In the past, there have been extensive studies on sparsity-driven un-supervised learning for denoising. *e.g.*, The KSVD method for dictionary learning [2] and data-driven wavelet frames [6]. Recently, There are also some works that train the network only on the target image itself, without calling any external training samples. The deep image prior (DIP) [30] uses early stopping to avoid overfitting, as it is observed that regular image patterns can be learned prior to random noise during the training. The Self2Self (S2S) method [26] adopts a dropout-based ensemble technique to handle the overfitting, which has the SOTA performance among existing single-image-based methods.

3. Main body

Recall that a noisy image $\boldsymbol{y}$ and its noise-free counterpart $\boldsymbol{x}$ is related by

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n},$$

where $\boldsymbol{n}$ denotes the random noise and follows the normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$. Typical supervised learning methods train the DNNs by

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}), \boldsymbol{x}), \tag{8}$$

where $\mathcal{L}(\cdot,\cdot)$ denotes some loss function and the squared ℓ_2-norm loss is used in the following. Without the access to clean images, simply replacing $\boldsymbol{x}$ in (8) with $\boldsymbol{y}$

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{y}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{y}\|_2^2, \tag{9}$$

will yields a trivial identity solution, i.e., the DNN does not remove any noise but outputs the noisy image itself.

Instead, for each noisy sample $\boldsymbol{y}$, our R2R training generates paired images $\{(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})\}$ as follows:

$$\widehat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{A}\boldsymbol{z}, \ \widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{B}\boldsymbol{z}, \tag{10}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ satisfies $\boldsymbol{A}\boldsymbol{B}^\top = \boldsymbol{\Sigma}_{\boldsymbol{x}}$ and $\boldsymbol{z}$ is sampled from standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then we train the DNN over $(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})$ by

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{A}, \boldsymbol{B}) := \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \boldsymbol{A}\boldsymbol{z}) - (\boldsymbol{y} - \boldsymbol{B}\boldsymbol{z})\|_2^2. \tag{11}$$

Denote $\widehat{\boldsymbol{n}} = \boldsymbol{n} + \boldsymbol{A}\boldsymbol{z}$ and $\widetilde{\boldsymbol{n}} = \boldsymbol{n} - \boldsymbol{B}\boldsymbol{z}$. It can be calculated that the covariance of $\widehat{\boldsymbol{n}}$ and $\widetilde{\boldsymbol{n}}$ is zero. Since they follow Gaussian distribution jointly, it yields that they are independent. Consequently, we have the following theorem regarding the loss function $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{A}, \boldsymbol{B})$ defined in (11).

Theorem 1. *Suppose $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ and $\boldsymbol{n}$ follows the normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$. Define a pair of images $(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})$ by (10), where $\boldsymbol{z}$ is independent from $\boldsymbol{n}$. Then with the condition $\boldsymbol{A}\boldsymbol{B}^\top = \boldsymbol{\Sigma}_{\boldsymbol{x}}$, it holds that*

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{A}, \boldsymbol{B}) = \widetilde{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{A}) + \text{const}, \tag{12}$$

where $\widetilde{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{A})$ is the supervised loss

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{A}) := \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \boldsymbol{A}\boldsymbol{z}) - \boldsymbol{x}\|_2^2. \tag{13}$$

Proof. See the supplemental material file for the proof. □

As an extension, we have the following corollary derived from Theorem 1.

Corollary 2. *Suppose $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ and $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$. The paired images $(\widehat{\boldsymbol{y}}, \widetilde{\boldsymbol{y}})$ are generated by*

$$\widehat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{D}^\top \boldsymbol{z}, \quad \widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{D}^{-1}\boldsymbol{z}, \tag{14}$$

where $\boldsymbol{z}$ draws from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ and is independent of $\boldsymbol{n}$. Then it holds that

$$\mathbb{E}_{\boldsymbol{y},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}) - \widetilde{\boldsymbol{y}}\|_2^2 = \mathbb{E}_{\widehat{\boldsymbol{y}}} \|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}) - \boldsymbol{x}\|_2^2 + \text{const}. \tag{15}$$

Theorem 1 implies that our R2R training is equivalent to training a denoiser over paired noisy/clean images $\{(\boldsymbol{y} + \boldsymbol{A}\boldsymbol{z}, \boldsymbol{x})\}$ in a supervised way. To test on the target noisy image $\boldsymbol{u}$, we feed $\boldsymbol{u} + \boldsymbol{A}\boldsymbol{z}$ into the trained DNN such that the statistics of the inputs are consistent during training and testing. Let $\boldsymbol{\theta}^*$ be the learned DNN parameter. The following scheme is used for prediction

$$\boldsymbol{u}^* = \int \mathcal{F}_{\boldsymbol{\theta}^*}(\boldsymbol{u} + \boldsymbol{A}\boldsymbol{z}) \Phi_{\boldsymbol{I}}(\boldsymbol{z}) d\boldsymbol{z} \approx \sum_{j=1}^{T} \mathcal{F}_{\boldsymbol{\theta}^*}(\boldsymbol{u} + \boldsymbol{A}\boldsymbol{z}^j), \tag{16}$$

where $\{\boldsymbol{z}^j\}_{j=1}^T$ are independent samples from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The Monte Carlo approximation is used to approximate the integration. The averaging of multiple forward processes is to reduce the effect of recorruption on the input image. However, if the DNN is trained over a sufficiently wide range of noise levels, the obtained R2R denoiser can also work well for the original noise level. In this case, we compute $\mathcal{F}_{\boldsymbol{\theta}^*}(\boldsymbol{u})$ directly to denoise the test image.

4. Experiments

In this section, we evaluate our proposed R2R training on AWGN denoising and real-world image denoising. More details can be found in the supplementary materials.

Our R2R training is independent of the network architectures. In our experiments, we use the same architecture as that of DnCNN [36], a baseline denoising DNN in the study of deep denoisers. The results of the compared methods are cited from the literature directly if possible. Otherwise, we use the pre-trained models or the codes provided by the authors to obtain the results. If none is available, *e.g*. Noisier2noise [23], we strictly follow the instructions of the paper to implement it by ourselves, and make efforts to optimize its performance.

Remark 3. *In the comparison, the best performer is emphasized by* **bold***, and the second best is colored in blue.*

4.1. Experiments on AWGN denoising

In this section, we test the denoising performance on the gray scale version of the BSD68 dataset which is corrupted by AGWN of two noise levels $\sigma = 25, 50$. The compared dataset-based learning methods are retrained on the benchmark denoising dataset BSD400 which contains 400 gray scale images of size 180×180. Noisy versions of all the images are generated by adding zero-mean white Gaussian noise with specific noise levels. For unsupervised methods, including N2V [17], N2S[5], SURE [29], Laine *et al*. [18], Nr2N [23] and our R2R, only noisy images are provided for training. For N2N [21], one more noisy version of each training image is generated. During training, the patches of size 40×40 are extracted from the training images and augmented by rotation, flipping and mirroring. For our method, a DnCNN network with 17 convolution layers is trained for 50 epochs with batch size 128. The initial learning rate is 10^{-3} and halves after 30 epochs. We generate our R2R image pairs for training by (14), where $\boldsymbol{D} = \alpha \boldsymbol{I}$ and $\boldsymbol{D}^{-1} = \boldsymbol{I}/\alpha$ with $\alpha = 0.5$. For prediction, we use the scheme (16) with $T = 50$.

See Tab. 1 for quantitative comparison of different methods on the testing dataset and Fig. 1 for visual comparison of some results. It can be seen that among all non-learning methods and unsupervised methods, the proposed R2R is the best performer in terms of both PSNR and SSIM. It is surprising to see that our method also outperformed N2N, which is weakly supervised on the noisy/noisy image pairs. On plausible cause might be that N2N can only utilize the provided noisy pairs while our method can generate multiple instances of image pairs from single noisy image, which makes our R2R generalize better. In comparison with representative supervised learning method DnCNN, the performance gap between our method and it is very small, less than 0.1dB in PSNR. That is, our proposed unsupervised method R2R is indeed comparable to its supervised counterpart, *i.e*. DnCNN.

4.2. Experiments on Real-Word Image Denoising

We test the performance of different methods on four real-world image datasets, *i.e*. CC [24], PloyU [33], SIDD Validation and SIDD Benchmark [1]. For CC, PolyU and SIDD Validation, ground truth images are provided. For SIDD Benchmark, the results are evaluated by submitting the denoised images to the project website [1]. The images in these datasets are captured by different cameras from different scenes and cropped to small image blocks for processing. There are 15 and 100 images of size 512×512 in the CC and PolyU dataset respectively. For both the SIDD Validation and Benchmark, images of 40 scenes are captured, each of which are cropped into 32 blocks of size 256×256, resulting in totally 1024 image blocks in the dataset.

SIDD dataset with unorganized noisy images for training. For SIDD, there is a training dataset with raw format available. The camera image processing pipeline is also available to convert the image in raw format to sRGB format. For noisy raw-RGB images in the training dataset, the noisy level function (NLF) is reported, which models the noise as a heteroscedastic signal dependent Gaussian variable with its variance proportional to the image intensity:

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \mathrm{diag}(\beta_1 \boldsymbol{x} + \beta_2), \tag{17}$$

where β_1 is the signal-dependent multiplicative component of the noise (the Poisson or shot noise), and β_2 is the independent additive Gaussian component of the noise.

We use the provided NLF to generate independent noisy raw-RGB image pairs by the scheme (14) with $\boldsymbol{D} = 2\boldsymbol{I}$, and $\boldsymbol{D}^{-1} = \boldsymbol{I}/2$, without calling the estimated clean images in the SIDD training dataset. These raw-RGB image pairs are then rendered to sRGB images using the provided camera image processing pipeline procedure for the following training of a sRGB-to-sRGB denoising DNN. Note that neither gamma correction nor tone mapping are called to generate the sRGB images provided in SIDD Validation and Benchmark dataset, and the same for our generated R2R sRGB noisy image pairs. As a result, the mean of image noise remains zero after being converted from raw-RGB space to sRGB space, and our method is still applicable. 320 noisy images in SIDD-Medium Dataset and a DnCNN with 20 convolution layers are used for our method. At each iteration, 32 pairs of image patches of size 128×128 from the dataset are extracted for training. The number of iteration is 5×10^5 and the learning rate is 5×10^{-5}. Here our R2R method is trained on the images with various noise levels, and the obtained denoising model is relatively insensitive to the noise level. Thus, there is no need to recorrupt the test images, *i.e*., we use the trained model $\mathcal{F}_{\boldsymbol{\theta}^*}(\cdot)$ directly for prediction during testing.

[1] https://www.eecs.yorku.ca/ kamel/sidd/

Table 1. Quantitative comparison, in PSNR(dB)/SSIM, of different methods for AWGN denoising on BSD68. The compared methods are categorized according to the type of training samples.

$\sigma = 25$	Single-image Based Methods				Noisy/Noisy	Noisy/Clean
	BM3D	WNMM	DIP	S2S	N2N	DnCNN
	28.56/0.801	28.80/0.809	27.96/0.774	28.57/0.802	28.86/0.823	**29.19/0.830**
	Trained on Unpaired Noisy Images					
	N2V	N2S	SURE	Nr2N	Laine *et al.*	R2R
	27.72/0.794	28.12/0.792	28.94/0.818	28.55/0.808	28.84/0.814	29.14/0.822
$\sigma = 50$	Single-image Based Methods				Noisy/Noisy	Noisy/Clean
	BM3D	WNMM	DIP	S2S	N2N	DnCNN
	25.62/0.687	25.87/0.698	25.04/0.645	25.93/0.698	25.77/0.700	**26.22/0.720**
	Trained on Unpaired Noisy Images					
	N2V	N2S	SURE	Nr2N	Laine *et al.*	R2R
	25.12/0.684	25.62/0.678	25.93/0.678	25.61/0.681	25.78/0.698	26.13/0.709

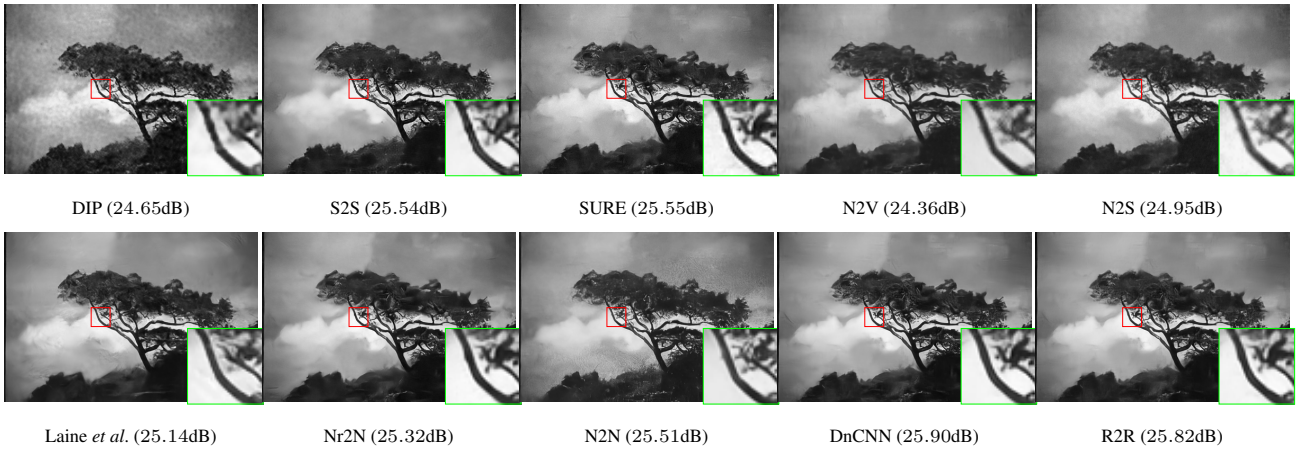


Figure 1. Visual results of removing AGWN of noise level $\sigma = 50$ on an example image from Set68.

In addition to two representative non-learning methods CBM3D and WNMM, two methods specifically designed for denoising real-world images are also included, namely multi-channel weighted nuclear norm minimization (MCWNNM) [34], and "noise clinic" (NC) method [19]. DnCNN, N2V and N2S are also retrained on SIDD-Medium for comparison, with (DnCNN) or without(N2V and N2S) calling the clean images in it. All the denoising methods are performed and evaluated on sRGB space.

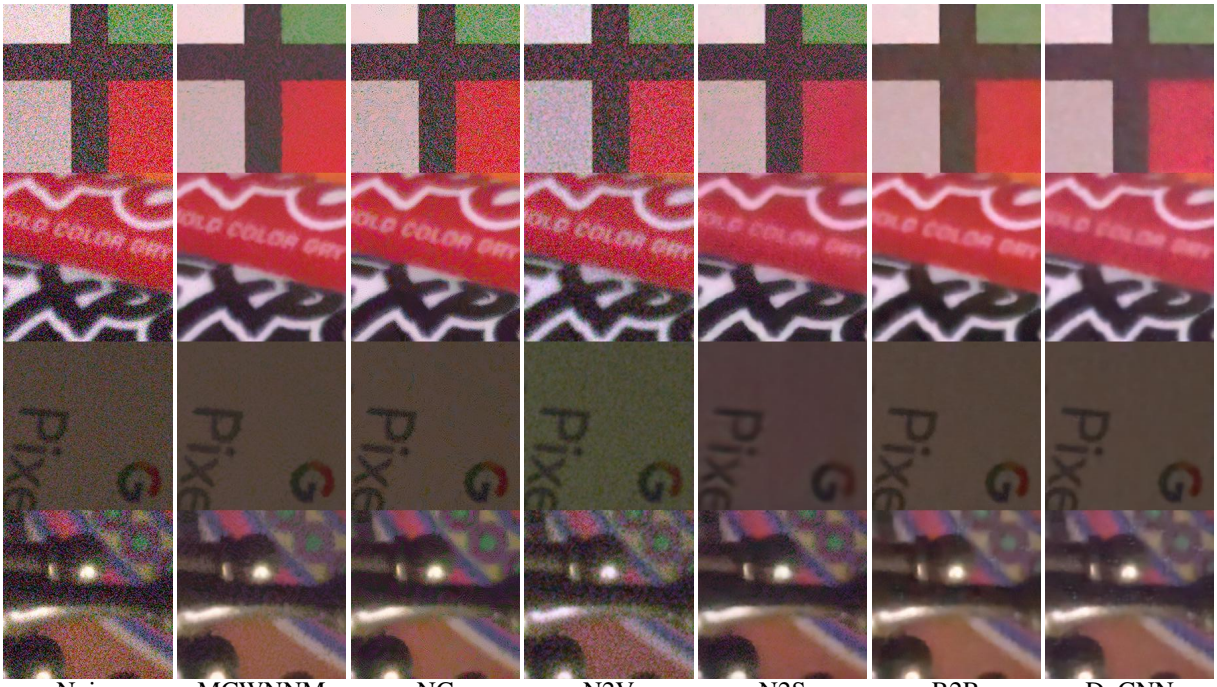
See Tab. 2 for quantitative comparison and Fig. 2 for visual comparison of some examples. It can be seen that the proposed R2R method outperformed all other non-learning methods and unsupervised methods. However, there is a noticeable performance gap between the R2R method and the supervised DnCNN, which may be caused by the inaccurate noise model and noise level function.

CC and PolyU dataset without external noisy training samples. For CC and PolyU, there is no training dataset available. Thus we train the denoiser on themselves directly without calling any external training samples. As noise characters are quite different for images captured under different conditions related to ISO level, shutter speed, illumination and other factors, we process these images individually. To obtain the results of DnCNN, we use the pre-trained blind DnCNN model for prediction, which are trained over the color version of BSD400 with AWGN where the noise level is uniformly sampled from $[0, 55]$.

For sRGB images in CC and PolyU, the noise model (17) is not applicable as the gamma correction and tone mapping in the camera image processing procedure distorted the statistical characters of noise from raw images. Thus, we simply model the noise by AWGN with different noise levels in different color channels, the same as MCWNNM [34]. The noise level is estimated using the method [9]. Then we set $\boldsymbol{A} = 20\sigma\boldsymbol{I}$ and $\boldsymbol{B} = \sigma\boldsymbol{I}/20$ in our recorruption scheme (10) for data generation with the estimated noise level σ for each color channel. Here a relative large recorruption coefficient 20 is used because the noise level in images from CC and PolyU is low and heavier recorruption is better for avoiding overfitting. For each image, we train a DnCNN with 17 convolution layers for 8000 iterations using a learning rate of 10^{-3}. It takes around half an hour to

Table 2. Quantitative comparison, in PSNR(dB)/SSIM, of different methods for denoising real-world images from SIDD.

Datasets	CBM3D	WNNM	MCWNNM	NC	N2V	N2S	R2R	DnCNN
SIDD Benchmark	25.65/0.685	25.78/0.809	33.37/0.875	31.26/0.826	27.68/0.668	29.56/0.808	34.78/0.898	**36.54/0.927**
SIDD Validation	25.65/0.475	26.20/0.693	33.40/0.815	31.31/0.725	29.35/0.651	30.72/0.787	35.04/0.844	**36.83/0.870**



Noisy MCWNNM NC N2V N2S R2R DnCNN

Figure 2. Visual results for denoising real-word images from SIDD Benchmark.

process one image of size $512 \times 512 \times 3$ for our method.

See Tab. 3 for the comparison of the R2R method to the pre-trained DnCNN and those methods that do not require any external training dataset, where N2V-single and N2S-single are the extensions of N2V and N2S to the case of single noisy image training. Our method still outperformed all other unsupervised deep learning methods by a large margin. There is a small advantages of our method over the top non-learning performer "MCWNNM" on the CC dataset, while a small disadvantages on the PolyU dataset. The pre-trained blind DnCNN performs poorly on CC and PolyU since it is trained on AWGN and generalizes poorly to the real noise. See Fig. 3 for visul comparison of some examples.

4.3. Ablation Study

This section is devoted to the ablation study for a better understanding of the proposed R2R method. We conduct the AWGN denoising experiments on BSD68 with noise level $\sigma = 25, 50$ in the following.

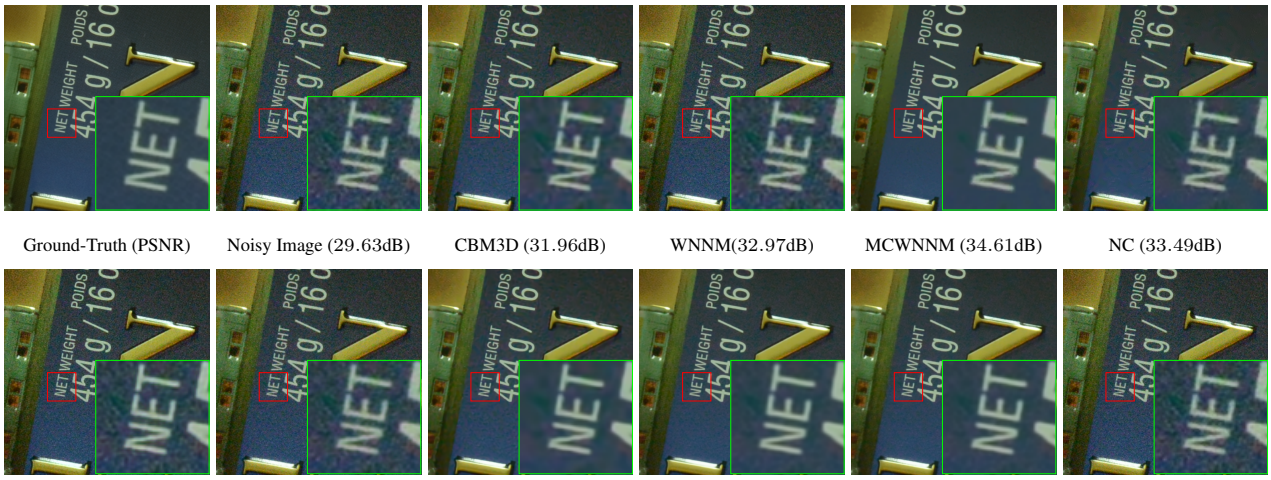
Performance gain from the prediction scheme (16). To show the benefit of the prediction scheme (16), we compare the result w/ it to the one w/o it. Tab. 4 shows that the performance gain brought by the scheme (16) is quite noticeable.

Performance impact of different value of α. Recall that we generate paired training data by (14) with $\boldsymbol{D} = \alpha \boldsymbol{I}$ for AWGN removal. To show the impact of the recorruption factor α on the performance, we compared the results yielded by using different values of α in the range $[0.1, 1]$. It can be seen from Tab. 5 that the impact of different values of α on the denoising performance is not significant.

Robustness to the estimation error of noise level. our method requires the prior knowledge on the noise levels of the training images to construct the pairs by (10). The sensitiveness of our method to the estimation error of noise level is evaluated. The experiments are conducted by contaminating the estimation of noise s.t.d. with up to 10% error, *i.e.* the noise level is sampled uniformly from $[0.9\sigma, 1.1\sigma]$ to generate recorrupted images. It can be seen from Tab. 6 that the impact of such error on the performance is negligible, which indicates the robustness of the proposed R2R method to the estimation error of noise level.

Table 3. Quantitative comparison, in PSNR(dB) /SSIM, of different methods for denoising real-world images from CC and PolyU.

Datasets	Methods				
CC	CBM3D	WNNM	MCWNNM	NC	DIP
	35.19/0.858	35.77/0.9381	37.70/**0.954**	36.43/0.936	37.37/0.947
	N2V-single	N2S-single	S2S	R2R-single	DnCNN
	32.27/0.862	33.38/0.846	37.52/0.947	**37.78**/0.951	33.47/0.932
PolyU	CBM3D	WNNM	MCWNNM	NC	DIP
	37.40/0.953	36.59/0.925	**38.51/0.967**	36.92/0.945	38.09/0.962
	N2V-single	N2S-single	S2S	R2R-single	DnCNN
	33.83/0.873	35.04/0.902	38.37/0.962	38.47/0.965	35.60/0.964



Ground-Truth (PSNR) Noisy Image (29.63dB) CBM3D (31.96dB) WNNM(32.97dB) MCWNNM (34.61dB) NC (33.49dB)

N2V-single (29.77dB) N2S-single (30.38dB) DIP (33.88dB) S2S (34.43dB) R2R (34.80dB) DnCNN(26.69dB)

Figure 3. Visual comparison of the results from different methods when denoising one image named as "d800-iso6400-1" from dataset CC.

Table 4. The PSNR (dB) gain from the prediction scheme (16).

Prediction	$\mathcal{F}_{\theta^*}(\boldsymbol{y})$	Our scheme (16)
$\sigma = 25$	28.89	**29.14**
$\sigma = 50$	25.86	**26.12**

Table 5. The impact of different values of α on the PSNR (dB).

α	1	0.5	0.3	0.1
$\sigma = 25$	28.81	**29.14**	29.03	28.98
$\sigma = 50$	25.81	**26.12**	25.93	25.74

Table 6. The robustness of the R2R method to the estimation error of noise level, in PSNR (dB).

estimation error of σ	10%	None
$\sigma = 25$	29.09	**29.14**
$\sigma = 50$	26.07	**26.12**

5. Conclusion

In this paper, we proposed an unsupervised deep learning denoising method trained on unpaired noisy images and proved that our training scheme has the same loss function as that of the supervised training up to a constant. It is further demonstrated by the numerical results on AWGN denoising, where our method is comparable to the supervised baseline. For both AWGN denoising and real-world image denoising, our method achieved the competitive results compared to the SOTA unsupervised learning methods.

Acknowledgment

Tongyao Pang, Huan Zheng and Hui Ji would like to acknowledge the support from Singapore MOE Academic Research Fund Tier 2 (Grant no. MOE2017-T2-2-156) and Tier 1 (Grant no. R-146-000-315-114). Yuhui Quan would like to acknowledge the support from National Natural Science Foundation of China (Grant No. 61872151) and CCF-Tencent Open Fund 2020.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proc. CVPR*, pages 1692–1700, 2018. 5

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006. 4

[3] Josue Anaya and Adrian Barbu. RENOIR–A dataset for real low-light image noise reduction. *J. Visual Comm. Image Representation*, 51:144–154, 2018. 1

[4] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *Proc. ICCV*, pages 3155–3164, 2019. 3

[5] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *Proc. ICML*, pages 524–533, 2019. 1, 3, 5

[6] Jian-Feng Cai, Hui Ji, Zuowei Shen, and Gui-Bo Ye. Data-driven tight frame construction and image denoising. *Applied and Computational Harmonic Analysis*, 37(1):89–105, 2014. 4

[7] Sungmin Cha, Taeeon Park, and Taesup Moon. Gan2gan: Generative noise learning for blind image denoising with single noisy images. *arXiv preprint arXiv:1905.10488*, 2019. 3

[8] Antonin Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1-2):89–97, 2004. 3

[9] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In *Proc. ICCV*, pages 477–485, 2015. 6

[10] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proc. CVPR*, pages 3155–3164, 2018. 3

[11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *Proc. ICIP*, pages 313–316, 2007. 2, 3

[12] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proc. CVPR*, pages 2862–2869, 2014. 3

[13] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proc. CVPR*, June 2019. 3

[14] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *Proc. CVPR*, pages 1791–1798. IEEE, 2010. 3

[15] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *Proc. CVPR*, pages 6054–6063, 2019. 3

[16] Rihuan Ke and Carola-Bibiane Schönlieb. Unsupervised image restoration using partially linear denoisers. *arXiv preprint arXiv:2008.06164*, 2020. 3

[17] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proc. CVPR*, pages 2129–2137, 2019. 1, 3, 5

[18] Samuli Laine, Jaakko Lehtinen, and Timo Aila. Self-supervised deep image denoising. *arXiv preprint arXiv:1901.10277*, 2019. 3, 5

[19] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Process. Line*, 5:1–54, 2015. 6

[20] Stamatios Lefkimmiatis. Universal denoising networks: a novel cnn architecture for image denoising. In *Proc. CVPR*, pages 3204–3213, 2018. 3

[21] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *Proc. ICML*, pages 2971–2980, 2018. 1, 2, 3, 5

[22] Christopher A Metzler, Ali Mousavi, Reinhard Heckel, and Richard G Baraniuk. Unsupervised learning with stein's unbiased risk estimator. *arXiv preprint arXiv:1805.10531*, 2018. 1

[23] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proc. CVPR*, pages 12064–12072, 2020. 1, 5

[24] 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 *Proc. CVPR*, pages 1683–1691, 2016. 5

[25] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proc. CVPR*, pages 1586–1595, 2017. 1

[26] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *Proc. CVPR*, pages 1890–1898, 2020. 1, 2, 4

[27] Yuhui Quan, Yixin Chen, Yizhen Shao, Huan Teng, Yong Xu, and Hui Ji. Image denoising using complex-valued deep cnn. *Pattern Recognition*, 111:107639, 2021. 3

[28] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 3

[29] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In *Proc. NeurIPS*, pages 3257–3267, 2018. 1, 2, 3, 5

[30] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proc. CVPR*, pages 9446–9454, 2018. 1, 4

[31] Raviteja Vemulapalli, Oncel Tuzel, and Ming-Yu Liu. Deep gaussian conditional random field network: A model-based deep network for discriminative denoising. In *Proc. CVPR*, pages 4801–4809, 2016. 1

[32] Jun Xu, Yuan Huang, Li Liu, Fan Zhu, Xingsong Hou, and Ling Shao. Noisy-as-clean: Learning unsupervised denoising from the corrupted image. *arXiv preprint arXiv:1906.06878*, 2019. 1

[33] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 1, 5

[34] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proc. ICCV*, pages 1096–1104, 2017. 6

[35] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. *arXiv preprint arXiv:2003.06792*, 2020. 3

[36] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017. 1, 2, 3, 5

[37] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.*, 27(9):4608–4622, 2018. 1, 3

Recorrupted-to-Recorrupted: Unsupervised Deep Learning for Image Denoising (Supplemental Materials)

Tongyao Pang[1], Huan Zheng[1], Yuhui Quan[2], and Hui Ji[1]
[1]Department of Mathematics, National University of Singapore, 119076, Singapore
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
matpt@nus.edu.sg, huan_zheng@u.nus.edu,csyhquan@scut.edu.cn, and matjh@nus.edu.sg

1. Proof of Theorem 1

Proof. Denote $\widehat{\boldsymbol{n}} = \boldsymbol{n} + \boldsymbol{A}\boldsymbol{z}$ and $\widetilde{\boldsymbol{n}} = \boldsymbol{n} - \boldsymbol{B}\boldsymbol{z}$. That is

$$\begin{pmatrix} \widehat{\boldsymbol{n}} \\ \widetilde{\boldsymbol{n}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{I} & -\boldsymbol{B} \end{pmatrix} \begin{pmatrix} \boldsymbol{n} \\ \boldsymbol{z} \end{pmatrix}. \tag{1}$$

Since $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and they are independent, we have

$$\begin{pmatrix} \widehat{\boldsymbol{n}} \\ \widetilde{\boldsymbol{n}} \end{pmatrix} \sim \mathcal{N}\Big(\boldsymbol{0}, \boldsymbol{\Sigma}'\Big), \tag{2}$$

where

$$\begin{aligned} \boldsymbol{\Sigma}' &= \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{I} & -\boldsymbol{B} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{A}^\top & -\boldsymbol{B}^\top \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{A}\boldsymbol{A}^\top & \boldsymbol{\Sigma}_{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{B}^\top \\ \boldsymbol{\Sigma}_{\boldsymbol{x}} - \boldsymbol{B}\boldsymbol{A}^\top & \boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{B}^\top \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{A}\boldsymbol{A}^\top & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{B}^\top \end{pmatrix}. \end{aligned} \tag{3}$$

Thus, $\widehat{\boldsymbol{n}}$ and $\widetilde{\boldsymbol{n}}$ are also independent Gaussian random variables. It yields that

$$\mathbb{E}_{\boldsymbol{x},\widehat{\boldsymbol{n}},\widetilde{\boldsymbol{n}}}\Big\{\widetilde{\boldsymbol{n}}^\top \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}})\Big\} = 0.$$

Then our loss function can be rewritten as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{A}, \boldsymbol{B}) =& \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \boldsymbol{A}\boldsymbol{z}) - (\boldsymbol{y} - \boldsymbol{B}\boldsymbol{z})\|_2^2 = \mathbb{E}_{\boldsymbol{x},\widehat{\boldsymbol{n}},\widetilde{\boldsymbol{n}}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) - (\boldsymbol{x} + \widetilde{\boldsymbol{n}})\|_2^2 \\ =& \mathbb{E}_{\boldsymbol{x},\widehat{\boldsymbol{n}},\widetilde{\boldsymbol{n}}}\Big\{ \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) - \boldsymbol{x}\|_2^2 - 2\widetilde{\boldsymbol{n}}^\top \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) + 2\widetilde{\boldsymbol{n}}^\top \boldsymbol{x} + \widetilde{\boldsymbol{n}}^\top \widetilde{\boldsymbol{n}} \Big\} \\ =& \mathbb{E}_{\boldsymbol{x},\widehat{\boldsymbol{n}},\widetilde{\boldsymbol{n}}}\Big\{ \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) - \boldsymbol{x}\|_2^2 + \widetilde{\boldsymbol{n}}^\top \widetilde{\boldsymbol{n}} \Big\} \\ =& \mathbb{E}_{\boldsymbol{x},\widehat{\boldsymbol{n}},\widetilde{\boldsymbol{n}}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x} + \widehat{\boldsymbol{n}}) - \boldsymbol{x}\|_2^2 + \mathbb{E}_{\boldsymbol{x}} \mathrm{trace}(\boldsymbol{\Sigma}_x + \boldsymbol{B}\boldsymbol{B}^\top) \\ =& \widetilde{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{A}) + \mathrm{const}. \end{aligned} \tag{4}$$

The proof is done. □

2. Running time

The inference time of processing the whole BSD68 dataset and SIDD Benchmark is around 115 seconds and 22 seconds respectively, on a NVIDIA TITAN RTX GPU with 24GB Memory. The reason why SIDD Benchmark is larger but takes less

time for inference is that the images in SIDD Benchmark are of the same size and can be processed in batch (a batch size of 32 is used by us), while the images in BSD68 vary in size and are processed one by one. Another cause is that, the AWGN denoiser is trained on the recorrupted images with specific noise level and thus the testing images in BSD68 are recorrupted for multiple times for prediction: $\sum_{j=1}^{50} \mathcal{F}_{\boldsymbol{\theta}^*}(\boldsymbol{u} + \boldsymbol{A}\boldsymbol{z}^j)$, while for SIDD Benchmark, the single forward prediction $\mathcal{F}_{\boldsymbol{\theta}^*}(\boldsymbol{u})$ is enough since the trained real-world image denoiser is blind to noise level.

3. Visual Comparison of More Examples

In this section, we provide visual comparison of more examples on AWGN denoising and real-world image denoising. See Fig. 1 – 6.

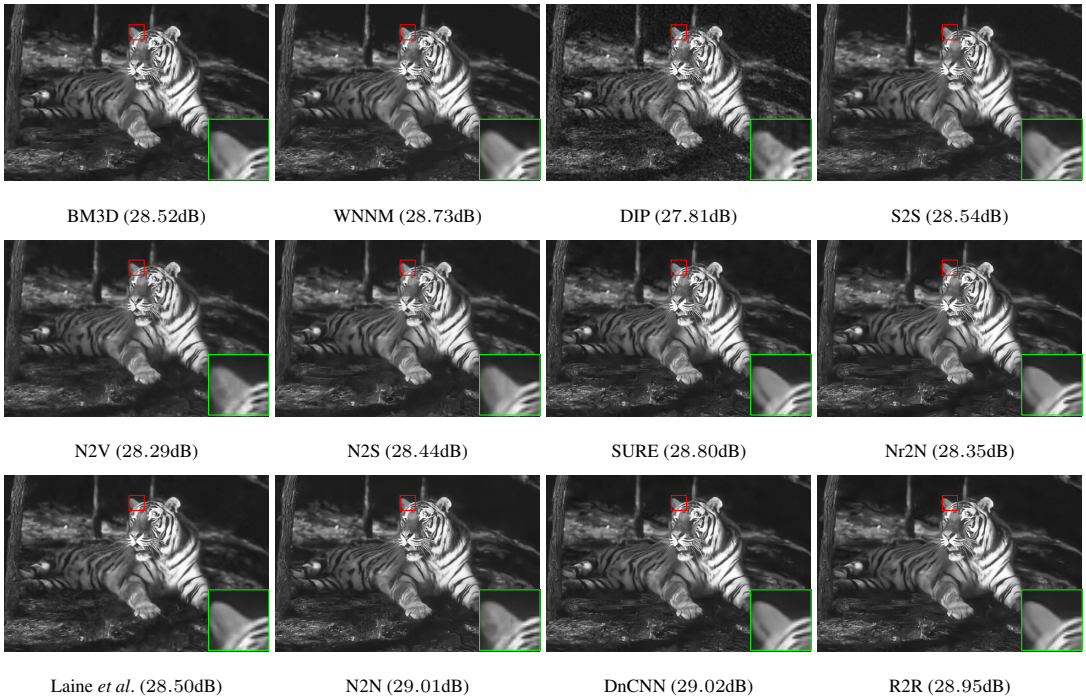
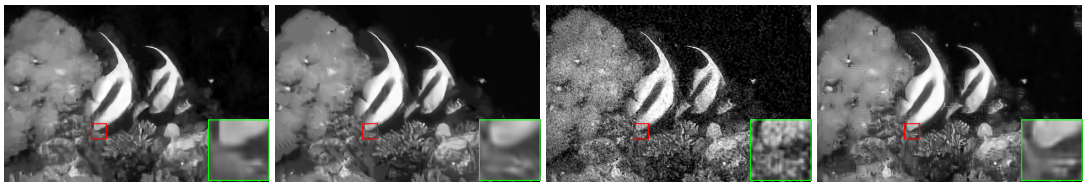
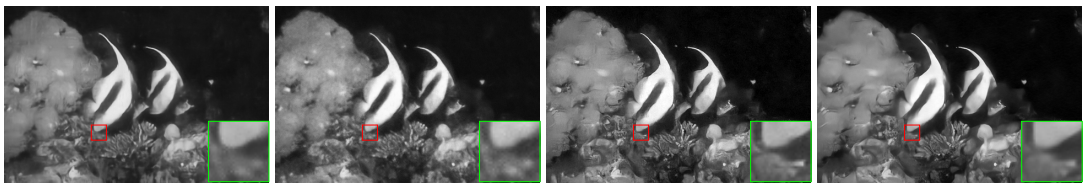


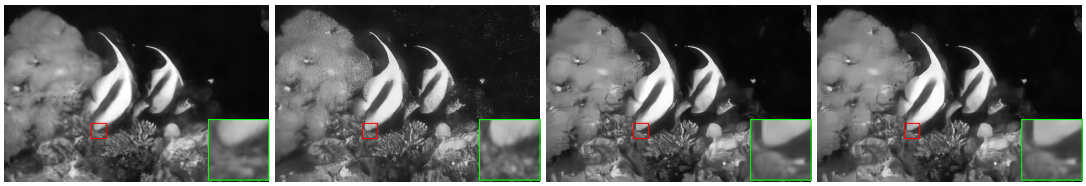
Figure 1. Visual results of removing AWGN of noise level $\sigma = 25$ on an example image from Set68.



BM3D (25.79dB) WNNM (25.91dB) DIP (22.16dB) S2S (25.85dB)

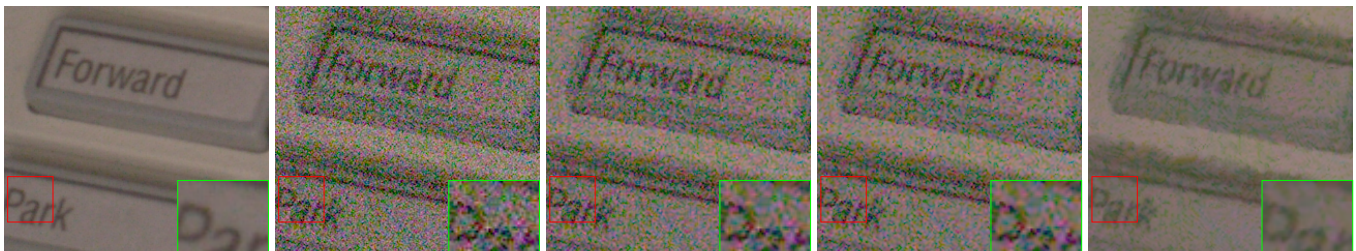


N2V (24.44dB) N2S (25.54dB) SURE (25.92dB) Nr2N (25.66dB)



Laine *et al*. (25.70dB) N2N (24.56dB) DnCNN (26.25dB) R2R (26.23dB)

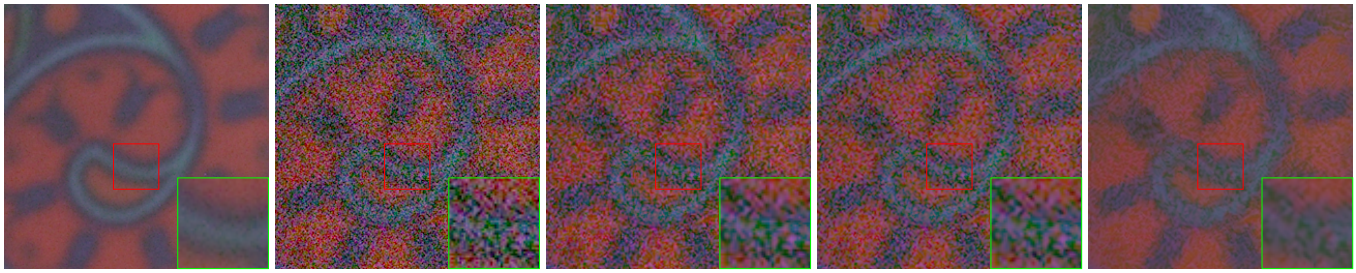
Figure 2. Visual results of removing AWGN of noise level $\sigma = 50$ on an example image from Set68.



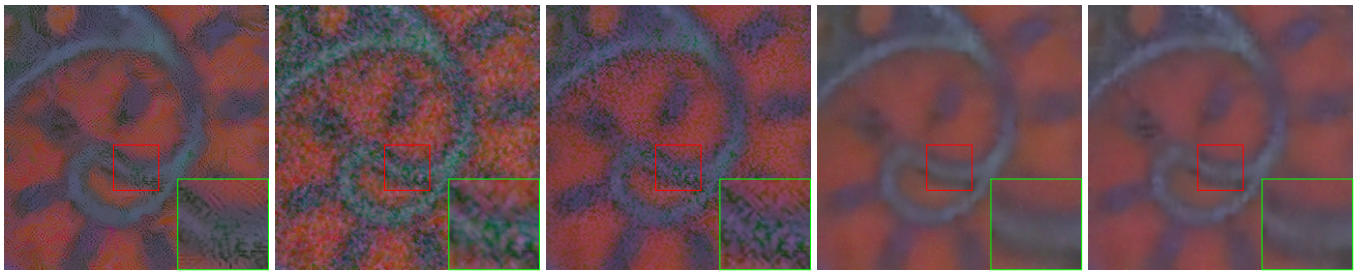
Truth (PSNR) Noisy (18.25dB) CBM3D (20.10dB) WNNM (20.71dB) MCWNNM (28.63dB)

NC (25.15dB) N2V (24.44dB) N2S (25.30dB) R2R (33.60dB) DnCNN (33.47dB)

Figure 3. Visual comparison of the results from different methods when denoising an example image from SIDD Validation.



Truth (PSNR) Noisy (17.68dB) CBM3D (20.04dB) WNNM (20.68dB) MCWNNM (27.72dB)



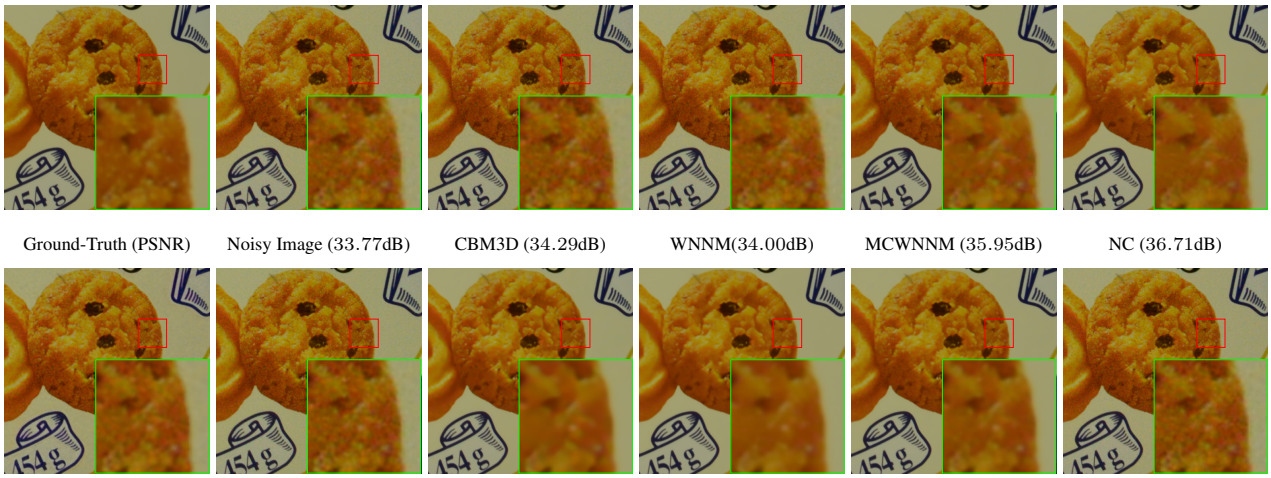
NC (26.22dB) N2V (24.00dB) N2S (25.34dB) R2R (30.37dB) DnCNN (31.17dB)

Figure 4. Visual comparison of the results from different methods when denoising an example image from SIDD Validation.

Noisy CBM3D WNNM MCWNNM TWSC

NC N2V N2S R2R DnCNN

Figure 5. Visual comparison of the results from different methods when denoising an example image from SIDD Benchmark.



Ground-Truth (PSNR) Noisy Image (33.77dB) CBM3D (34.29dB) WNNM(34.00dB) MCWNNM (35.95dB) NC (36.71dB)

N2V-single (33.02dB) N2S-single (33.98dB) DIP (37.44dB) S2S (37.22dB) R2R (38.01dB) DnCNN(34.21dB)

Figure 6. Visual comparison of the results from different methods when denoising an example image from dataset CC.