Self-supervised Image Denoising with Downsampled Invariance Loss and Conditional Blind-Spot Network

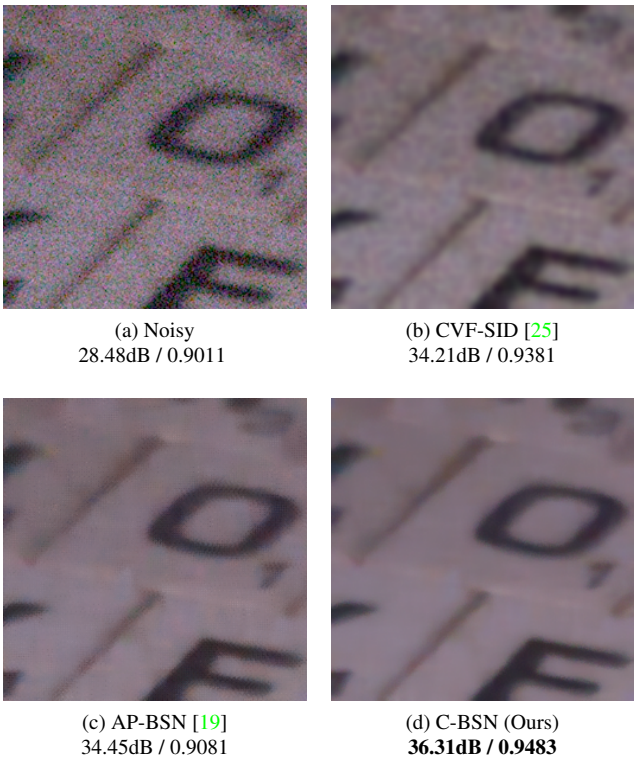
Yeong Il Jang[1] Keuntek Lee[1] Gu Yong Park[1] Seyun Kim[2] Nam Ik Cho[1,3]
[1]Department of ECE, INMC, Seoul National University, Seoul, Korea
[2]Gauss Labs Inc.
[3]IPAI, Seoul National University, Seoul, Korea
{jyicu, leekt000, pgy9134}@snu.ac.kr, seyun.kim@gausslabs.ai, nicho@snu.ac.kr

Abstract

There have been many image denoisers using deep neural networks, which outperform conventional model-based methods by large margins. Recently, self-supervised methods have attracted attention because constructing a large real noise dataset for supervised training is an enormous burden. The most representative self-supervised denoisers are based on blind-spot networks, which exclude the receptive field's center pixel. However, excluding any input pixel is abandoning some information, especially when the input pixel at the corresponding output position is excluded. In addition, a standard blind-spot network fails to reduce real camera noise due to the pixel-wise correlation of noise, though it successfully removes independently distributed synthetic noise. Hence, to realize a more practical denoiser, we propose a novel self-supervised training framework that can remove real noise. For this, we derive the theoretic upper bound of a supervised loss where the network is guided by the downsampled blinded output. Also, we design a conditional blind-spot network (C-BSN), which selectively controls the blindness of the network to use the center pixel information. Furthermore, we exploit a random subsampler to decorrelate noise spatially, making the C-BSN free of visual artifacts that were often seen in downsample-based methods. Extensive experiments show that the proposed C-BSN achieves state-of-the-art performance on real-world datasets as a self-supervised denoiser and shows qualitatively pleasing results without any post-processing or refinement.

1. Introduction

Image denoising aims to recover a clean image from its corrupted counterpart. Recently, image denoisers using convolutional neural networks (CNNs) have achieved great performances, significantly outperforming conventional model-based ones [43, 44, 32]. They trained networks by minimizing the difference between the network outputs and the ground-truth clean images. In early works, they assumed the camera noise as an additive white Gaussian noise (AWGN) and generated a large number of clean-noisy image pairs for the supervised training. However, the denoisers trained with AWGN fail to generalize to real-world camera noises due to the difference between the Gaussian and real noise distributions [10]. Specifically, real



(a) Noisy
28.48dB / 0.9011

(b) CVF-SID [25]
34.21dB / 0.9381

(c) AP-BSN [19]
34.45dB / 0.9081

(d) C-BSN (Ours)
36.31dB / 0.9483

Figure 1. **Visual comparison of denoised images on SIDD validation [2].** Our C-BSN shows better details and no artifacts without post-processing or refinement. Best viewed in pdf.

noise follows a more complicated distribution than a simple Gaussian and gets more correlated spatially and chromatically while passing through an in-camera image processing pipeline, such as demosaicing that involves the computation using adjacent pixels.

Some researchers attempted to find a more realistic noise model to deal with real noise. In the case of camera-raw images, noise can be modeled with a relatively simple distribution such as heteroscedastic Gaussian [8]. Hence, a raw image added with such synthetic noise is passed through a camera image signal processor (ISP) model to generate a realistic noisy sRGB image [10, 40]. Other works synthesized realistic noise using generative models [6, 5, 12, 1]. Another approach is to construct paired real noise datasets from real photos like DND [27] and SIDD [2]. Training in a supervised manner with those datasets successfully reduced the noise of real cameras [3, 41, 42]. However, acquiring aligned clean images corresponding to noisy ones requires a series of static photos of the same scene. It is costly or even impossible in some cases, such as medical images, since it requires strictly controlled capturing and complicated post-processing. Also, since they used several cameras in specific environments for capturing real noises, they might have different distributions from the ones captured from other cameras and from the same cameras with different shooting environments.

To mitigate the necessity of large aligned datasets, self-supervised denoising that requires only noisy images has been proposed. The most representative methods are based on blind-spot networks (BSN), where each output pixel is estimated from the surrounding noisy pixels except for the corresponding one. It enables the network to learn with the self-supervised loss function, where the same noisy images are used as both input and target. The idea of blind-spot prevents the network from converging to a trivial identity function. The BSN is shown to converge to the clean image under the assumption that the expectation of the noise is zero and the noise is pixel-wise independent. They imposed blindness to the network by masking the input image [17, 4] or by designing networks that structurally exclude the central pixel from the receptive fields [18, 35, 19]. However, the BSN-based self-supervised algorithms have two limitations; 1) The network cannot utilize the center pixel which is the most informative. 2) It is not applicable to real noise since it has a pixel-wise correlation in the sRGB domain [19].

In this paper, we propose a novel self-supervised learning framework to denoise real noise without the blind-spot, *i.e.*, with the center pixel information. Our framework overcomes the above-stated limitations by deriving a novel downsampled invariance loss function. The downsampled invariance loss employs a novel conditional blind-spot network (C-BSN) and random subsampler. Specifically, our C-BSN conditionally controls its blindness by switching the masked convolution operations. It allows the network to be regularized by its blind-spot counterpart, which prevents the trivial solution. Furthermore, we impose the loss on randomly downsampled subimage so that the correlation of the noise is weakened without inducing visual artifacts. In addition, we augment the loss with a blind self-supervised loss for stabilizing the training. Extensive experiments have been conducted to evaluate the proposed framework, which validates that the C-BSN outperforms existing self-supervised denoisers and even some supervised methods trained with real noise datasets.

The contributions of our method are summarized as follows:

- We propose a novel self-supervised denoising framework that can be processed without a blind-spot. We theoretically derive the upper bound of the self-supervised loss as downsampled invariance loss, which exploits masked output as the regularization of the denoised image without masking. In addition, the proposed method does not require post-processing or noise statistics.
- To apply downsampled invariance loss, we propose a novel conditional blind-spot network named C-BSN, which conditionally controls the blindness of the network. To deal with the spatial correlation of the real camera noise, a random subsampler is proposed to avoid visual artifacts.
- The C-BSN shows state-of-the-art performance in real-world sRGB benchmarks DND [27] and SIDD [2], as shown in Figs. 1, 4, and 5.

2. Related Works

Deep Image Denoising Image denoisers based on Convolutional Neural Networks (CNNs) have outperformed conventional model-based algorithms. In early works, deep image denoisers were trained with large datasets consisting of clean images and noisy ones corrupted by synthetic Gaussian noise. DnCNN [43] proposed a CNN denoiser with batch normalization and residual learning. Following DnCNN, many networks with more sophisticated architectures have been proposed [44, 23, 32, 21, 45]. However, denoisers trained with synthetic Gaussian noise could not generalize well for denoising real-world noisy images. To alleviate this problem, CBDNet [10] synthesized heteroscedastic Gaussian noise and processed it through the camera ISP model. Some works simulated realistic noise using generative adversarial network (GAN) [6, 5, 12] or flow-based methods [1, 22, 16]. With the development of real-world sRGB datasets [2, 27], recent denoisers have been trained and tested on these datasets, [3, 41, 42, 39, 13, 14, 31, 33],

demonstrating that the real noisy images could be successfully denoised. Moreover, it has been shown that earlier denoisers can also work better by retraining with these datasets. However, collecting a large dataset is laborious and costly. Moreover, the networks trained with a specific dataset may not function properly on images captured by other cameras, not included in the dataset, or images from other domains, such as medical, electron, and ultra-sonic.

Self-supervised Deep Image Denoising In order to overcome the lack of aligned real noisy-clean image pairs, self-supervised learning that trains denoiser with solely noisy images has been proposed. Lehtinen *et al*. [20] proposed Noise2Noise where training pairs are two noisy images of the same scene. Noise2Void [17] and Noise2Self [4] introduced self-supervised denoisers that require only single noisy images by masking the center pixel of the receptive field. Without masking input pixels, Laine *et al*. [18] proposed a structurally blind-spotted network with a concatenation of half-plane receptive field U-Nets [29]. Wu *et al*. [35] introduced dilated blind-spot network (D-BSN), where masked convolution is followed by dilated convolutions and 1×1 convolutions, strictly excluding the center pixel from the receptive field. Self2Self [28] trained the denoiser with a single noisy image by applying Bernoulli dropout. Neighbor2Neighbor [11] proposed a self-supervised loss between two subsampled images. Also, assuming known noise characteristics, Noisy-as-clean [37] and Noisier2noise [24] added a proper noise to the noisy image and used the pair as a training set. Recorrupted2Recorrupted [26] generated pairs of Gaussian-corrupted images to be used as training pairs. In general, real noises of the sRGB domain have unknown or non-stationary statistics and are spatially correlated, making the above methods less applicable.

Recently, some works have been proposed to overcome the limitations of the above BSN-based methods. To mitigate the spatial correlation of real noise, AP-BSN [19] utilized pixel downshuffle (PD) [46] asymmetrically. During training, the network was trained using high strides where the assumption of independence holds. During testing, low strides were used to preserve more pixel information. CVF-SID [25] disentangled a clean image and signal-dependent noise from real-world noisy input. To utilize information of center pixel, Laine *et al*. [18] post-processed the denoised output to be the posterior with the known noise model in a Bayesian approach. Noise2Same [36] derived the upper bound of self-supervised loss without introducing the blind-spot. Blind2Unblind [34] proposed re-visible loss that makes blind-spot visible again. However, to the best of our knowledge, there has been no research that handles both problems (use of blind-spot and handling spatial correlation) for self-supervised image denoising.

3. Method

3.1. Overview

We introduce a novel self-supervised learning framework to denoise real-world RGB images, which is illustrated in Fig. 2. We propose a novel loss function that can be directly optimized on the input image without loss of information. It consists of self-supervised loss and downsampled invariance loss that controls the extent of the blindness. Our main idea of the downsampled invariance loss is to make a blind-spot network serve as regularization of the same network while preserving network parameters. To this end, we propose a conditional blind-spot network (denoted C-BSN in the figure) to selectively mask the center pixel in the receptive field. In addition, we introduce Random Subsampler (RS) to decorrelate noise spatially. The pixel-shuffle downsampling (PD) [46] also loosens the spatial correlation of the noise, but it generates severe checkerboard artifacts. On the contrary, since our RS draws a pixel randomly from each grid, it does not produce such artifacts. We denote the noisy input image as $\boldsymbol{x}$ and the corresponding clean image as $\boldsymbol{y}$. For brevity, the channel dimension is omitted, and spatial dimensions are vectorized, *i.e*., $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$.

3.2. Revisiting Noise2Same

Under the assumption that noise is zero mean and pixel-wise independent, Baston *et al*. [4] proved that self-supervised loss is equivalent to supervised loss if the network is $\mathcal{J}$-invariant.

Definition 1. *[4] Let $\mathcal{J}$ be a partition of the dimensions $\{1, ..., m\}$ and let $J \subset \mathcal{J}$. A function $f : \mathbb{R}^m \to \mathbb{R}^m$ is J-invariant if $f(\boldsymbol{x})_J$ does not depend on the value of $\boldsymbol{x}_J$. It is $\mathcal{J}$-invariant if it is J-invariant for each $J \in \mathcal{J}$.*

Subscripted notation $\boldsymbol{x}_J$ is used for $\boldsymbol{x}$ restricted to J. Noise2Same [36] analyzed that strictly $\mathcal{J}$-invariant function is not optimal for the denoisers. Rather, it mitigates the $\mathcal{J}$-invariance constraints by minimizing the upper bound of supervised loss,

$$\begin{aligned}\mathcal{L}_{N2Same} = \mathbb{E}_{\boldsymbol{x}} ||f(\boldsymbol{x}) - \boldsymbol{x}||^2/m \\ + \lambda_{inv}\, \mathbb{E}_J(\mathbb{E}_{\boldsymbol{x}} ||f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^C})_J||^2/|J|)^{\frac{1}{2}},\end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the normalized input image so that the mean of $\boldsymbol{x}$ is zero and the standard deviation equals one. The first term is the self-supervised loss, while the second term controls how $\mathcal{J}$-invariant f should be.

3.3. Downsampled Invariance Loss

Noise2Same upper bound holds when $f(\boldsymbol{x}_{J^C})$ in Eq. (1) is not correlated with $\boldsymbol{x}_J$. Although the pixel-wise independent noise such as AWGN satisfies the above constraint, real noise is correlated spatially, which makes it no longer

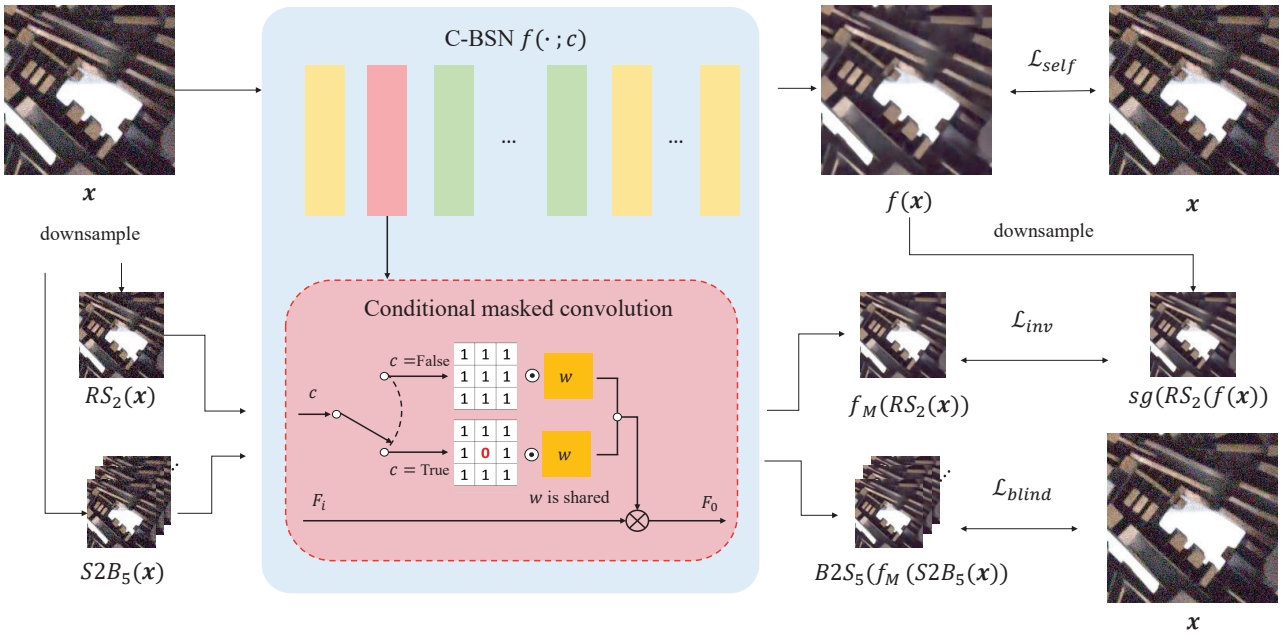


Figure 2. **Overview of the proposed C-BSN framework.** Illustration of the C-BSN architecture and loss functions. For simplicity, condition variable c is omitted in f when c = False, and f_M denotes the blind-spot network with c = True. The yellow box represents 1×1 convolution and the green box represents dilated convolution module, which consists of dilated convolution followed by 1×1 convolution and residual skip connection. Note that RS samples the same pixel indices when calculating downsampled invariance loss with $RS_2(f(\boldsymbol{x}))$ and $sg(f_M(RS_2(\boldsymbol{x})))$.

applicable. Instead of randomly sampling the subset J, we sample the downsampled image to reduce the correlation, following previous research [46, 19]. Precisely, we propose modified version of Eq. (1) as follows:

Proposition 1. *Let $\boldsymbol{x}$ be a normalized zero-mean noisy image conditioned on $\boldsymbol{y}$, $\mathbb{E}[\boldsymbol{x}|\boldsymbol{y}] = \boldsymbol{y}$. Let d be any downsampling operation and $d_s(\boldsymbol{x})$ be a set of downsampled pixels of $\boldsymbol{x}$ with a stride of s. Assume that downsampled subimage $d_s(\boldsymbol{x})$ has zero pixel-wise correlation and f_M is a blind-spot network. Then, the following inequality holds.*

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\left\|f(\boldsymbol{x})-\boldsymbol{y}\right\|^2 + \|\boldsymbol{x}-\boldsymbol{y}\|^2 \leq \mathbb{E}_{\boldsymbol{x}}\left\|f(\boldsymbol{x})-\boldsymbol{x}\right\|^2 + 2\sqrt{ms^2}\,\mathop{\mathbb{E}}_{d_s(\boldsymbol{x})}[\mathbb{E}\left\|d_s(f(\boldsymbol{x}))-f_M(d_s(\boldsymbol{x}))\right\|^2]^{\frac{1}{2}}. \quad (2)$$

Proposition 1 provides the upper bound of the supervised loss with the self-supervised loss and the regularization of the downsampled output with the blind output of the downsampled input. We prove in the supplementary material that $f(\boldsymbol{x}_{J^C})$ in Eq. (1) can be replaced by $f_M(d_s(\boldsymbol{x}))$, which has no correlation with $d_s(\boldsymbol{x})$. This simplifies the second term of Eq. (1) to our new downsampled invariance loss,

$$\mathcal{L}_{inv} = \sqrt{\frac{s^2}{m}}\left\|d_s(f(\boldsymbol{x})) - sg(f_M(d_s(\boldsymbol{x})))\right\|_2, \quad (3)$$

where sg is a stop-gradient operation. With Proposition 1, we can optimize the denoising network by minimizing the right side of Eq. (2). Details of the proof are in the supplementary material.

3.4. Conditional Blind-Spot Network

Equation (3) requires the parameters of the network f to be shared regardless of the blind-spot. In the case of Noise2Same [36], the network remains unchanged as blindness is caused by masking input pixels, not by the network structure. However, masking causes train-test discrepancy of inputs and harms training efficiency since loss can be back-propagated only through masked pixels. On the other hand, a network such as D-BSN [35] excludes the center pixel by its architecture. It can be optimized through every single pixel, though the blindness cannot be removed. To control blindness with D-BSN architecture conditionally, the network structure should be changed while sharing the training parameters. To this end, we propose a conditional blind-spot network (C-BSN) to make a blind-spot without masking the input image.

In D-BSN, blindness is induced by masked convolutions, and dilated convolutions prevent masked pixel information from being mixed in. We switch the behavior of masked convolution by changing the mask of kernels according to the given condition c:

$$F_o = (M \odot W) * F_i + b, \quad (4)$$

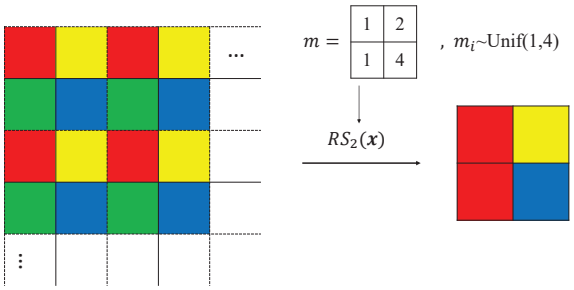


Figure 3. **Details of a random subsampler with a stride of two.** Each color represents a relative position within the cell. m is a selection mask introduced for an explanation. The indices of m that determine which pixel will be selected are randomly sampled from the uniform distribution.

$$M = \begin{cases} \mathbf{1}_{k\times k} - \boldsymbol{\delta}_{k\times k}, & \text{if } c = \text{True}, \\ \mathbf{1}_{k\times k}, & \text{otherwise}, \end{cases} \tag{5}$$

where W is a convolutional filter, b is a bias, and F_i and F_o are input features and output features, respectively. $\boldsymbol{\delta}_{k\times k}$ is a $k \times k$ Dirac delta kernel, and $\mathbf{1}$ is the matrix of ones. For simplicity, we omit the condition variable c when $c =$ False and represent only blind-conditioned network as $f_M = f(\cdot; c = \text{True})$. We only use f_M in the training phase, and all test images are inferred by non-blind network f without the loss of information.

Applying conditional masked convolution can alter the output features' distribution because the kernel's center is set to zero when c is False. However, f should be trained differently from f_M to utilize the masked pixel. In addition, the center of the kernel is trained independently of f_M, based on the modified feature distribution. Hence, we use the kernel and its mask without normalization between $c =$True and $c =$False.

3.5. Random Subsampler

In Section 3.3, we introduced a downsample operator to the invariance loss to extract a subset of the image with zero spatial correlation. This constraint is guaranteed in pixel-wise synthetic noises, while real noise does not comply. In order to remove spatial pixel dependency, Zhou *et al*. [46] and Lee *et al*. [19] utilized pixel-shuffle downsampling (PD). The PD is the inverse operation of the pixel-shuffle [30] and creates the mosaic of the subimages. However, directly applying PD in downsampled invariance loss is not trivial since the expectation of Eq. (2) is calculated over subimages $d_s(\boldsymbol{x})$. Another approach to decorrelate the noise is a space2batch (S2B) operation, where pixel down-shuffled subimages are concatenated along batch dimension instead of channel dimension. However, naively applying S2B induces severe visual artifacts in the results. When S2B images are taken as input, all the subimages are calculated independently, which results in a checkerboard pattern in the batch2space (B2S) upsampled outputs, giving false guidance to the $f(x)$.

To deal with this problem, we propose a random subsampler $RS_s(\cdot)$, a subsampling operator to avoid the checkerboard artifact. Figure 3 shows the details of our random subsampler. Taking stride of two as an example, input images are divided into 2×2 grid cells. For each cell, a pixel is randomly drawn within the cell, making s times downsampled image. If the randomly downsampled pixel in the adjacent cell is also adjacent, the correlation may occur significantly. However, in this case, the average distance from the other peripheral pixels becomes large, and the expected average distance between subsampled pixels can still be approximated to s. Therefore, as with PD, the expected spatial correlation is weakened by the random subsampler.

3.6. Total Loss function

In this section, we provide the total loss function. For simple notation, we use $\|\cdot\|$ to represent the pixel-averaged L_1 norm. We substitute mean squared errors to the L_1 norm in the self-supervised loss, as

$$\mathcal{L}_{self} = \left\| f(x) - x \right\|. \tag{6}$$

Also, we find it beneficial to replace the root mean square (RMS) of the downsampled invariance loss with the L_1 norm as well and to use a random subsampler as a downsampling operation,

$$\mathcal{L}_{invRS} = \left\| RS_2(f(\boldsymbol{x})) - sg(f_M(RS_2(\boldsymbol{x}))) \right\|. \tag{7}$$

From the Proposition 1 in Section 3.3, we minimize the upper bound of supervised loss function,

$$\mathcal{L}_{CBSN} = \mathcal{L}_{self} + \lambda_{inv} \cdot \mathcal{L}_{invRS} \tag{8}$$

where λ_{inv} is a hyperparameter to control the contribution of the downsampled invariance loss. We set the stride of RS as 2 in order to reflect more spatial information.

In addition, we introduce a self-supervised loss of the blind conditioned network, $\mathcal{L}_{blind}$, to stabilize the training as in [34], where

$$\mathcal{L}_{blind} = \left\| B2S_5(f_M(S2B_5(\boldsymbol{x}))) - \boldsymbol{x} \right\|. \tag{9}$$

While downsampled invariance loss utilizes the stride of two, the stride in Eq. (9) is five since the ideal BSN should be trained with as little correlation as possible. Without blind self-supervised loss, $f_M(\boldsymbol{x})$ is random in the early stage of training, giving wrong guidance to the $f(\boldsymbol{x})$. Thus, we augment $\mathcal{L}_{CBSN}$ with the blind self-supervised loss to facilitate the transition from f_M to f. Additionally, we adopt warm-up scheduling to $\mathcal{L}_{CBSN}$. Scheduling parameter λ_{sch} is multiplied to $\mathcal{L}_{CBSN}$, gradually increasing the

Table 1. **Quantitative comparison on SIDD and DND benchmarks.** PSNR and SSIM are from the official SIDD and DND websites. We use † notation to indicate that the network is trained on the test set directly. $*$ denotes that the method uses a self-ensemble strategy. The highest PSNR and SSIM of self-supervised algorithms are highlighted in **bold**.

Supervision	Method	SIDD		DND	
		PSNR(dB)	SSIM	PSNR(dB)	SSIM
Model-based	BM3D [7]	25.65	0.685	34.51	0.851
	WNNM [9]	25.78	0.809	34.67	0.865
Supervised	DNCNN [43]	35.13	0.896	37.89	0.932
	CBDNet [10]	33.28	0.868	38.05	0.942
	RIDNet [3]	38.70	0.950	39.24	0.952
	AINDNet (R)* [14]	38.84	0.951	39.34	0.952
	VDN [39]	39.26	0.955	39.38	0.952
	MIRNet [41]	39.72	0.959	39.88	0.956
	MAXIM-3S [33]	39.96	0.960	39.84	0.957
Generation-based	GCBD [37]	-	-	35.58	0.922
	C2N* [12] + DIDN [38]	35.35	0.937	36.38	0.887
Self-supervised	NAC [37]	-	-	36.20	0.925
	R2R [26]	34.78	0.898	-	-
	CVF-SID(T) [25]	34.43	0.912	36.31	0.923
	CVF-SID(S^2)† [25]	34.71	0.917	36.50	0.924
	AP-BSN [19]	34.90	0.900	37.46	0.924
	AP-BSN + R^3 [19]	35.97	0.925	38.09	0.937
	C-BSN	36.82	**0.934**	38.45	0.939
	C-BSN†	**36.84**	0.933	**38.60**	**0.941**

impact of $\mathcal{L}_{CBSN}$. With all these in consideration, the total objective function is defined as

$$\mathcal{L}_{total} = \mathcal{L}_{blind} + \lambda_{sch} \cdot \mathcal{L}_{CBSN}. \tag{10}$$

4. Experimental results

4.1. Implementation Details

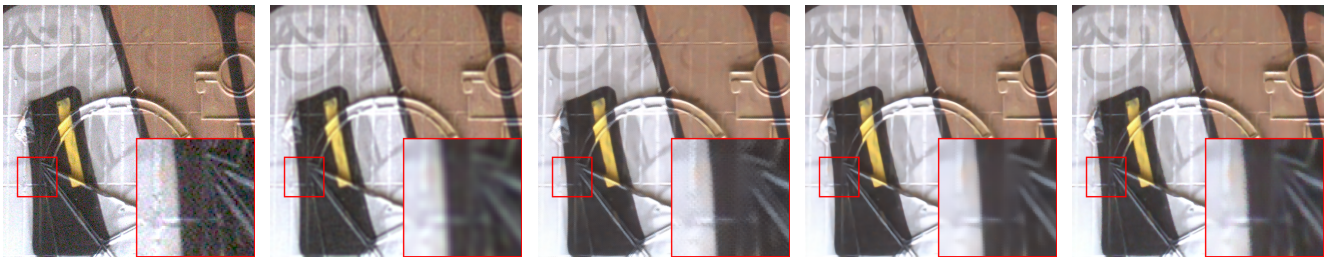
We train and test our method on real-world sRGB camera noise. Our model is trained in two settings; one is trained with an external dataset, and the other is trained with a test set directly. For the external training set, we use the SIDD medium set [2], which contains 320 pairs of aligned real noisy-clean images captured by five smartphone cameras. We only use the noisy images as training samples and discard all clean images. In addition, as C-BSN requires only noisy images to be trained, we train C-BSN† solely on test set images. We test the proposed algorithm in DND [27] and SIDD [2] benchmark. DND consists of 50 high-resolution noisy images from four different cameras. Note that both benchmarks evaluate PSNR and SSIM online and do not provide ground truth images.

We crop 240×240 patches from training images and use the mini-batch size of 4. We randomly rotate 90° and flip for data augmentation for each image patch. Input images are normalized so that the mean and the standard deviation are 0 and 1, respectively. The standard deviation is calculated as $\max(\text{std}, \frac{1}{\sqrt{m}})$ to avoid division by zero.

We follow the AP-BSN structure [19] with modified masked convolution in order to compare the effectiveness of loss functions only. We set λ_{inv} to 2 as derived in Proposition 1 and employ a warm-up strategy for λ_{sch} that linearly increases from 0 to 1 for the first 200,000 iterations. We use Adam [15] optimizer with the initial learning rate 1e-4. C-BSN is optimized for 400,000 iterations, and the learning rate is halved every 100,000 iterations, capped at 2e-5. Note that C-BSN requires a single inference of input image, and the downsampling operation is not performed in test time.

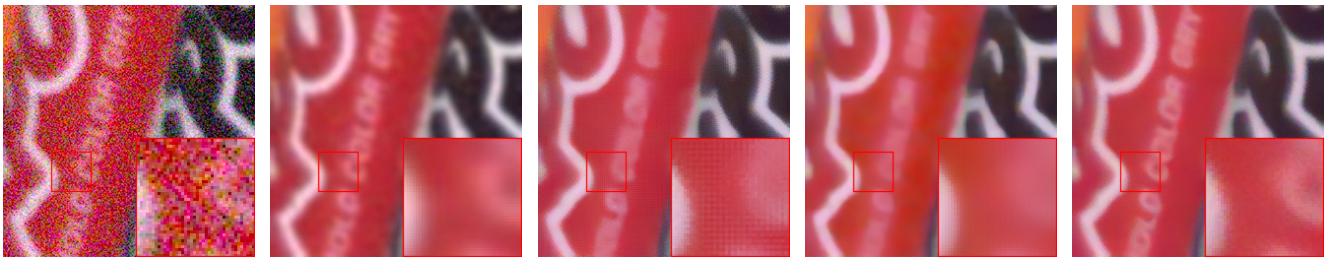
4.2. Comparison with state-of-the-art algorithms

We compare our C-BSN against supervised, generation-based, and self-supervised methods. The supervised models are trained on real noisy-clean pairs of SIDD, and the generation-based models simulate realistic noise and train denoiser with generated pairs. The self-supervised models use only noisy images to train the networks. We only report the self-supervised models that aim to remove the real noise. Table 1 compares PSNR and SSIM on SIDD and DND benchmarks. The proposed C-BSN outperforms other



(a) Noisy image 26.90 / 0.7527 | (b) CVF-SID(S^2) [25] 28.74/0.8737 | (c) AP-BSN [19] 30.79 / 0.8901 | (d) AP-BSN+R^3 [19] 32.04 / 0.9164 | (e) C-BSN† **32.63 / 9180**

Figure 4. **Visual comparison on DND benchmark.** PSNR and SSIM of each image are reported below.



(a) Noisy image | (b) CVF-SID(S^2) [25] | (c) AP-BSN [19] | (d) AP-BSN+R^3 [19] | (e) C-BSN†

Figure 5. **Visual comparison on SIDD benchmark.** In SIDD benchmark, PSNR and SSIM of the image is not available.

self-supervised methods by large margins and even some supervised networks. C-BSN† trained with the test dataset shows slightly higher PSNR than the C-BSN trained on the external dataset. It demonstrates that the training with the same noise distribution of the test set benefits the performance of the network. Specifically, C-BSN† outperforms CVF-SID (S^2) and AP-BSN+R^3 by 2.13dB and 0.51dB, which shows the effectiveness of our framework. The proposed downsampled invariance loss and C-BSN structure enjoy the use of blind-spot information and single inference with full image resolution.

Figs. 1, 4, and 5 illustrate the qualitative comparisons of self-supervised methods on the DND and SIDD benchmarks. We can see that the outputs of CVF-SID remain noisy and show stains in the flat region. AP-BSN suffers from checkerboard artifact and AP-BSN+R^3 over-blur image details. On the contrary, it can be seen that our C-BSN successfully reduces the noise and preserves the structure of the images.

Note that AP-BSN+R^3 [19] and CVF-SID(S^2) [25] exploit a refinement technique that requires multiple runs of the network. AP-BSN+R^3 randomly replaces denoised pixels with noisy ones and averages the denoised results of randomly replaced inputs. CVF-SID(S^2) trains the second model with the denoised images as a new training set and double-denoise with two successive models. On the other hand, we do not need any post-processing and achieve state-of-the-art results with a single inference.

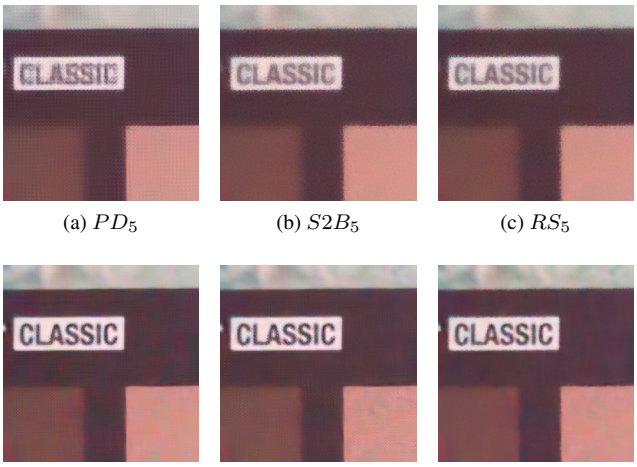
Table 2. **Ablation on loss function.** Details of the settings of the experiment are reported in Section 4.3.

Loss function	PSNR(dB)	SSIM
$\mathcal{L}_{N2Same}$	25.58	0.807
$\mathcal{L}_{total}$ with blind-spot	35.86	0.931
$\mathcal{L}_{inv}$ with RMS	35.63	0.920
$\mathcal{L}_{total}$	**36.22**	**0.935**

4.3. Ablation Study

In this section, we conduct ablation studies on the loss function, downsampler, and blind loss to show the effectiveness of the proposed method. To reduce the cost of training, we train the networks with the patch size of 120 $\times$ 120 and evaluate them on the SIDD validation set.

Ablation on loss function. We analyze the different loss functions to evaluate the effectiveness of our downsampled invariance loss and conditional blind-spot network. Table 2 reports the PSNR on the SIDD validation dataset with four different loss functions. For $\mathcal{L}_{N2Same}$, we set all condition c to False so that the network is not blind, and the blindness is caused by masking input as in [36]. The network trained with $\mathcal{L}_{N2Same}$ fails to converge, showing that a downsampling operation is necessary to reduce the spatial correlation of real noisy input. $\mathcal{L}_{total}$ with blind-spot is trained with original D-BSN, which is not able to remove the blind-spot. We set all c to True to make the network blind while keeping the other loss functions the same. Note that it differs from AP-BSN or D-BSN since the network is trained by $\mathcal{L}_{total}$ on



(a) PD_5 (b) $S2B_5$ (c) RS_5

(d) PD_2 (e) $S2B_2$ (f) RS_2

Figure 6. **Qualitative comparison of different downsampling operations in downsampled invariance loss on SIDD validation.**

full image resolution. We can see that PSNR drops largely without C-BSN structure, which validates the importance of the center pixel information. Lastly, $\mathcal{L}_{inv}$ with L_2 is trained by $\mathcal{L}_{invRS}$ with the RMS as in Noise2Same. The performance decreases when the L_1 norm of $\mathcal{L}_{invRS}$ is replaced by RMS, which shows L1 norm can enhance the quality of output significantly.

Ablation on downsampler. We evaluate the networks trained with different downsamplers in the downsampled invariance loss to validate the effectiveness of our random subsampler with a stride of two. We test three downsamplers, PD, S2B, and RS, with strides of 2 and 5. Each stride represents the small stride for more information and the large stride for spatial independence of real noise. Table 3 and **??** show the effectiveness of each downsample operation quantitatively and qualitatively. As argued in Section 3.5, the networks trained with PD underperform S2B and produce visual artifacts of size $s \times s$. The models trained with the stride of 5 produce blurry images and cannot remove noise around the edges. It demonstrates that it is advantageous to keep spatial information of the input with a small stride in the downsampled invariance loss. Regardless of the stride, S2B outperforms PD, and RS outperforms S2B. PD and S2B with a stride of two can reduce spatially correlated noise, but it also produces severe checkerboard artifacts. On the other hand, the proposed RS_2 achieves the highest PSNR and visually pleasing result without artifacts, outperforming PD_2 and $S2B_2$ by 0.90dB and 0.20dB, respectively.

Ablation on the blind loss. We investigate the effectiveness of the blind loss, $\mathcal{L}_{blind}$. Though $\mathcal{L}_{CBSN}$ is an upper bound of the supervised loss, the training is unstable without $\mathcal{L}_{blind}$. We set the hyperparameter λ_{sch} to different conditions as in Table 4. When $\lambda_{sch} = \infty$, we do not use the blind loss and train C-BSN with $\mathcal{L}_{CBSN}$ only. In this case, the network fails to learn denoising and outputs zeros, resulting in a flat image of the input mean. With $\lambda_{sch} = 0$, the loss function is $\mathcal{L}_{blind}$ as AP-BSN [19]. However, processing AP-BSN with the original size input without a blind-spot produces severe artifacts and poor image quality. It can be seen that $\lambda_{sch} = 1$ shows suboptimal PSNR to warm-up, yet it sometimes falls to the same local optima as $\lambda_{sch} = \infty$. The suggested warm-up scheduling brings about 0.57dB PSNR improvement and stabilizes the training procedure.

Table 3. **Ablation on the downsampler of downsampled invariance loss.**

downsampler	stride	PSNR(dB)	SSIM
PD	5	34.71	0.905
	2	35.32	0.914
$S2B$	5	35.62	0.924
	2	36.02	0.922
RS	5	35.24	0.922
	2	**36.22**	**0.935**

Table 4. **Ablation on the blind loss.**

λ_{sch}	PSNR(dB)	SSIM
∞	25.92	0.810
0	29.59	0.757
1	35.65	0.926
warm-up	**36.22**	**0.935**

5. Conclusion

We have presented a novel self-supervised image denoising framework C-BSN for real camera noise reduction. We have derived the downsampled invariance loss, which is the upper bound of the supervised loss and enables the training without a blind-spot. The C-BSN structure conditionally controls blind-spot, and then the random subsampler decorrelates noise without introducing visual artifacts. Without using post-processing or refinement, our C-BSN outperforms recent self-supervised denoisers. Our code is available at https://github.com/jyicu/CBSN.

Acknowledgement

This research was supported in part by Samsung Electronics Co., Ltd., in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (2021R1A2C2007220), and partially by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-01062).

References

[1] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3165–3173, 2019. 2

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 1, 2, 6

[3] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3155–3164, 2019. 2, 6

[4] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019. 2, 3

[5] Yuanhao Cai, Xiaowan Hu, Haoqian Wang, Yulun Zhang, Hanspeter Pfister, and Donglai Wei. Learning to generate realistic noisy images via pixel-level noise-aware adversarial training. *Advances in Neural Information Processing Systems*, 34:3259–3270, 2021. 2

[6] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3155–3164, 2018. 2

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

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

[9] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014. 6

[10] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1712–1722, 2019. 1, 2, 6

[11] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14781–14790, 2021. 3

[12] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2n: Practical generative noise modeling for real-world denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2350–2359, 2021. 2, 6

[13] Yeong Il Jang, Yoonsik Kim, and Nam Ik Cho. Dual path denoising network for real photographic noise. *IEEE Signal Processing Letters*, 27:860–864, 2020. 2

[14] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3482–3492, 2020. 2, 6

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 6

[16] Shayan Kousha, Ali Maleky, Michael S Brown, and Marcus A Brubaker. Modeling srgb camera noise with normalizing flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17463–17471, 2022. 2

[17] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2129–2137, 2019. 2, 3

[18] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 3

[19] Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17725–17734, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[20] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018. 3

[21] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. *Advances in neural information processing systems*, 31, 2018. 2

[22] Ali Maleky, Shayan Kousha, Michael S Brown, and Marcus A Brubaker. Noise2noiseflow: Realistic camera noise modeling without clean images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17632–17641, 2022. 2

[23] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29, 2016. 2

[24] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12064–12072, 2020. 3

[25] Reyhaneh Neshatavar, Mohsen Yavartanoo, Sanghyun Son, and Kyoung Mu Lee. Cvf-sid: Cyclic multi-variate function for self-supervised image denoising by disentangling noise from image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17583–17591, 2022. 1, 3, 6, 7

[26] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: unsupervised deep learning for

image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2043–2052, 2021. 3, 6

[27] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017. 2, 6

[28] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1890–1898, 2020. 3

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[30] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 5

[31] Jae Woong Soh and Nam Ik Cho. Variational deep image restoration. *IEEE Transactions on Image Processing*, 31:4363–4376, 2022. 2

[32] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017. 1, 2

[33] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022. 2, 6

[34] Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2unblind: Self-supervised image denoising with visible blind spots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2027–2036, 2022. 3, 5

[35] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *European conference on computer vision*, pages 352–368. Springer, 2020. 2, 3, 4

[36] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. Noise2same: Optimizing a self-supervised bound for image denoising. *Advances in Neural Information Processing Systems*, 33:20320–20330, 2020. 3, 4, 7

[37] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE Transactions on Image Processing*, 29:9316–9329, 2020. 3, 6

[38] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 6

[39] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. *Advances in neural information processing systems*, 32, 2019. 2, 6

[40] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020. 2

[41] 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. In *European Conference on Computer Vision*, pages 492–511. Springer, 2020. 2, 6

[42] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 2

[43] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 1, 2, 6

[44] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1, 2

[45] Y Zhang, K Li, B Zhong, and Y Fu. Residual non-local attention networks for image restoration. In *International Conference on Learning Representations*, 2019. 2

[46] Yuqian Zhou, Jianbo Jiao, Haibin Huang, Yang Wang, Jue Wang, Honghui Shi, and Thomas Huang. When awgn-based denoiser meets real noises. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13074–13081, 2020. 3, 4, 5