Unpaired Learning of Deep Image Denoising

Xiaohe Wu[1], Ming Liu[1], Yue Cao[1], Dongwei Ren[2], and Wangmeng Zuo[1,3] (✉)

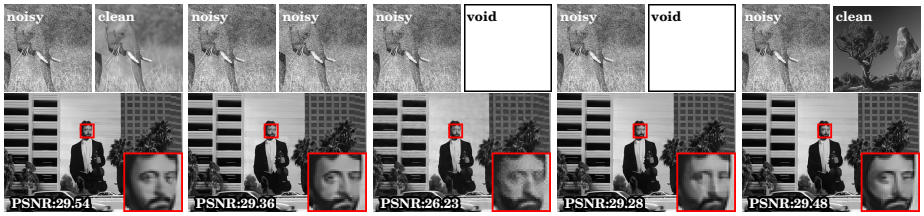
[1] Harbin Institute of Technology, China
[2] University of Tianjin, China
[3] Peng Cheng Lab, China
csxhwu@gmail.com, csmliu@outlook.com,
{cscaoyue,rendongweihit}@gmail.com, wmzuo@hit.edu.com

Abstract. We investigate the task of learning blind image denoising networks from an unpaired set of clean and noisy images. Such problem setting generally is practical and valuable considering that it is feasible to collect unpaired noisy and clean images in most real-world applications. And we further assume that the noise can be signal dependent but is spatially uncorrelated. In order to facilitate unpaired learning of denoising network, this paper presents a two-stage scheme by incorporating self-supervised learning and knowledge distillation. For self-supervised learning, we suggest a dilated blind-spot network (D-BSN) to learn denoising solely from real noisy images. Due to the spatial independence of noise, we adopt a network by stacking 1×1 convolution layers to estimate the noise level map for each image. Both the D-BSN and image-specific noise model (CNN_{est}) can be jointly trained via maximizing the constrained log-likelihood. Given the output of D-BSN and estimated noise level map, improved denoising performance can be further obtained based on the Bayes' rule. As for knowledge distillation, we first apply the learned noise models to clean images to synthesize a paired set of training images, and use the real noisy images and the corresponding denoising results in the first stage to form another paired set. Then, the ultimate denoising model can be distilled by training an existing denoising network using these two paired sets. Experiments show that our unpaired learning method performs favorably on both synthetic noisy images and real-world noisy photographs in terms of quantitative and qualitative evaluation. Code is available at https://github.com/XHWXD/DBSN.

Keywords: Image denoising, unpaired learning, convolutional networks, self-supervised learning

1 Introduction

Recent years have witnessed the unprecedented success of deep convolutional neural networks (CNNs) in image denoising. For additive white Gaussian noise (AWGN), numerous CNN denoisers, e.g., DnCNN [40], RED30 [30], MWCNN [27], N3Net [34], and NLRN [26], have been presented and achieved noteworthy improvement in denoising performance against traditional methods such as



MWCNN(N2C) [27] MWCNN(N2N) [25] N2V [22] Laine19 [24] MWCNN(unpaired)

Fig. 1: Supervision settings for CNN denoisers, including Noise2Clean (MWCNN(N2C) [27]), Noise2Noise [25] (MWCNN(N2N)), Noise2Void (N2V [22]), Self-supervised learning (Laine19 [24]), and our unpaired learning scheme.

BM3D [9] and WNNM [14]. Subsequently, attempts have been made to apply CNN denoisers for handling more sophisticated types of image noise [18, 35] as well as removing noise from real-world noisy photographs [2, 6, 8, 15].

Albeit breakthrough performance has been achieved, the success of most existing CNN denoisers heavily depend on supervised learning with large amount of paired noisy-clean images [2, 6, 15, 30, 38, 40, 42]. On the one hand, given the form and parameters of noise model, one can synthesize noisy images from noiseless clean images to constitute a paired training set. However, real noise usually is complex, and the in-camera signal processing (ISP) pipeline in real-world photography further increases the complexity of noise, making it difficult to be fully characterized by basic parametric noise model. On the other hand, one can build the paired set by designing suitable approaches to acquire the nearly noise-free (or clean) image corresponding to a given real noisy image. For real-world photography, nearly noise-free images can be acquired by averaging multiple noisy images [2, 31] or by aligning and post-processing low ISO images [33]. Unfortunately, the nearly noise-free images may suffer from over-smoothing issue and are cost-expensive to acquire. Moreover, such nearly noise-free image acquisition may not be applicable to other imaging mechanisms (e.g., microscopy or medical imaging), making it yet a challenging problem for acquiring noisy-clean image pairs for other imaging mechanisms.

Instead of supervised learning with paired training set, Lehtinen et al. [25] suggest a Noise2Noise (N2N) model to learn the mapping from pairs of noisy instances. However, it requires that the underlying clean images in each pair are exactly the same and the noises are independently drawn from the same distribution, thereby limiting its practicability. Recently, Krull et al. [22] introduce a practically more feasible Noise2Void (N2V) model which adopts a blind-spot network (BSN) to learn CNN denoisers solely from noisy images. Unfortunately, BSN is computationally very inefficient in training and fails to exploit the pixel value at blind spot, giving rise to degraded denoising performance (See Fig. 1). Subsequently, self-supervised model [24] and probabilistic N2V [23] have been further suggested to improve training efficiency via masked convolution [24] and to improve denoising performance via probabilistic inference [23, 24].

N2V [22] and self-supervised model [24], however, fail to exploit clean images in training. Nonetheless, albeit it is difficult to acquire the nearly noise-free

image corresponding to a given noisy image, it is practically feasible to collect a set of unpaired clean images. Moreover, specially designed BSN architecture generally is required to facilitate self-supervised learning, and cannot employ the progress in state-of-the-art networks [26,27,30,34,38,40–42] to improve denoising performance. Chen et al. [8] suggest an unpaired learning based blind denoising method GCBD based on the generative adversarial network (GAN) [17], but only achieve limited performance on real-world noisy photographs.

In this paper, we present a two-stage scheme, i.e., self-supervised learning and knowledge distillation, to learn blind image denoising network from an unpaired set of clean and noisy images. Instead of GAN-based unpaired learning [7,8], we first exploit only the noisy images to learn a BSN as well as an image-specific noise level estimation network CNN_{est} for image denoising and noise modeling. Then, the learned noise models are applied to clean images for synthesizing a paired set of training images, and we also use the real noisy images and the corresponding denoising results in the first stage to form another paired set. As for knowledge distillation, we simply train a state-of-the-art CNN denoiser, e.g., MWCNN [27], using the above two paired sets.

In particular, the clean image is assumed to be spatially correlated, making it feasible to exploit the BSN architecture for learning blind denoising network solely from noisy images. To improve the training efficiency, we present a novel dilated BSN (i.e., D-BSN) leveraging dilated convolution and fully convolutional network (FCN), allowing to predict the denoising result of all pixels with a single forward pass during training. We further assume that the noise is pixel-wise independent but can be signal dependent. Hence, the noise level of a pixel can be either a constant or only depends on the individual pixel value. Considering that the noise model and parameters may vary with different images, we suggest an image-specific CNN_{est} by stacking 1×1 convolution layers to meet the above requirements. Using unorganized collections of noisy images, both D-BSN and CNN_{est} can be jointly trained via maximizing the constrained log-likelihood. Given the outputs of D-BSN and CNN_{est}, we use the Bayes' rule to obtain the denoising result in the first stage. As for a given clean image in the second stage, an image-specific CNN_{est} is randomly selected to synthesize a noisy image.

Extensive experiments are conducted to evaluate our D-BSN and unpaired learning method, e.g., MWCNN(unpaired). On various types of synthetic noise (e.g., AWGN, heteroscedastic Gaussian, multivariate Gaussian), our D-BSN is efficient in training and is effective in image denoising and noise modeling. While our MWCNN(unpaired) performs better than the self-supervised model Laine19 [24], and on par with the fully-supervised counterpart (e.g., MWCNN [27]) (See Fig. 1). Experiments on real-world noisy photographs further validate the effectiveness of our MWCNN(unpaired). As for real-world noisy photographs, due to the effect of demosaicking, the noise violates the pixel-wise independent noise assumption, and we simply train our blind-spot network on pixel-shuffle down-sampled noisy images to circumvent this issue. The results show that our MWCNN(unpaired) also performs well and significantly surpasses GAN-based unpaired learning GCBD [8] on DND [33].

The contributions are summarized as follows:

1. A novel two-stage scheme by incorporating self-supervised learning and knowledge distillation is presented to learn blind image denoising network from an unpaired set of clean and noisy images. In particular, self-supervised learning is adopted for image denoising and noise modeling, consequently resulting in two complementary paired set to distill the ultimate denoising network.
2. A novel dilated blind-spot network (D-BSN) and an image-specific noise level estimation network CNN_{est} are elaborated to improve the training efficiency and to meet the assumed noise characteristics. Using unorganized collections of noisy images, D-BSN and CNN_{est} can be jointly trained via maximizing the constrained log-likelihood.
3. Experiments on various types of synthetic noise show that our unpaired learning method performs better than N2V [22] and Laine19 [24], and on par with its fully-supervised counterpart. MWCNN(unpaired) also performs well on real-world photographs and significantly surpasses GAN-based unpaired learning (GCBD) [8] on the DND [33] dataset.

2 Related Work

2.1 Deep Image Denoising

In the last few years, significant progress has been made in developing deep CNN denoisers. Zhang et al. [40] suggested DnCNN by incorporating residual learning and batch normalization, and achieved superior performance than most traditional methods for AWGN removal. Subsequently, numerous building modules have been introduced to deep denoising networks, such as dilated convolution [41], channel attention [4], memory block [38], and wavelet transform [27]. To fulfill the aim of image denoising, researchers also modified the representative network architectures, e.g., U-Net [27,30], Residual learning [40], and non-local network [26], and also introduced several new ones [19, 34]. For handling AWGN with different noise levels and spatially variant noise, FFDNet was proposed by taking both noise level map and noisy image as network input. All these studies have consistently improved the denoising performance of deep networks [4,19,26,27,30,34,38,41]. However, CNN denoisers for AWGN usually generalize poorly to complex noise, especially real-world noisy images [33].

Besides AWGN, complex noise models, e.g., heteroscedastic Gaussian [33] and Gaussian-Poisson [11], have also been suggested, but are still not sufficient to approximate real sensor noise. As for real-world photography, the introduction of ISP pipeline makes the noise both signal-dependent and spatially correlated, further increasing the complexity of noise model [13, 28]. Instead of noise modeling, several methods have been proposed to acquire nearly noise-free images by averaging multiple noisy images [2,31] or by aligning and post-processing low ISO images [33]. With the set of noisy-clean image pairs, several well-designed CNNs were developed to learn a direct mapping for removing noise from real-world noisy RAW and sRGB images [6,15,39]. Based upon the GLOW architecture [20],

Abdelhamed et al. [1] introduced a deep compact noise model, i.e., Noise Flow, to characterize the real noise distribution from noisy-clean image pairs. In this work, we learn a FCN with 1×1 convolution from noisy images for modeling pixel-independent signal-dependent noise. To exploit the progress in CNN denoising, we further train state-of-the-art CNN denoiser using the synthetic noisy images generated with the learned noise model.

2.2 Learning CNN Denoisers without Paired Noisy-Clean Images

Soltanayev and Chun [37] developed a Steins unbiased risk estimator (SURE) based method on noisy images. Zhussip et al. [44] further extended SURE to learn CNN denoisers from correlated pairs of noisy images. But the above methods only handle AWGN and require that noise level is known.

Lehtinen et al. [25] suggested to learn an N2N model from a training set of noisy image pairs, which avoids the acquisition of nearly noise-free images but remains limited in practice. Subsequently, N2V [22] (Noise2Self [5]) has been proposed to learn (calibrate) denoisers by requiring the output at a position does not depend on the input value at the same position. However, N2V [22] is inefficient in training and fails to exploit the pixel value at blind spot. To address these issues, Laine19 [24] and probabilistic N2V [23] were then suggested by introducing masked convolution [24] and probabilistic inference [23, 24].

N2V and follow-up methods are solely based on noisy images without exploiting unpaired clean images. Chen et.al [8] presented a GAN-based model, i.e., GCBD, to learn CNN denoisers using unpaired noisy and clean images, but its performance on real-world noisy photographs still falls behind. In contrast to [8], we develop a non-GAN based method for unpaired learning. Our method involves two stage in training, i.e., self-supervised learning and knowledge distillation. As opposed to [22–24], our self-supervised learning method elaborately incorporates dilated convolution with FCN to design a D-BSN for improving training efficiency. For modeling pixel-independent signal-dependent noise, we adopt an image-specific FCN with 1×1 convolution. And constrained log-likelihood is then introduced to train D-BSN and image-specific CNN_{est}.

3 Proposed Method

In this section, we present our two-stage training scheme, i.e., self-supervised learning and knowledge distillation, for learning CNN denoisers from an unpaired set of noisy and clean images. After describing the problem setting and assumptions, we first introduce the main modules of our scheme and explain the knowledge distillation stage in details. Then, we turn to self-supervised learning by describing the dilated blind-spot network (D-BSN), image-specific noise model CNN_{est} and self-supervised loss. For handling real-world noisy photographs, we finally introduce a pixel-shuffle down-sampling strategy to apply our method.

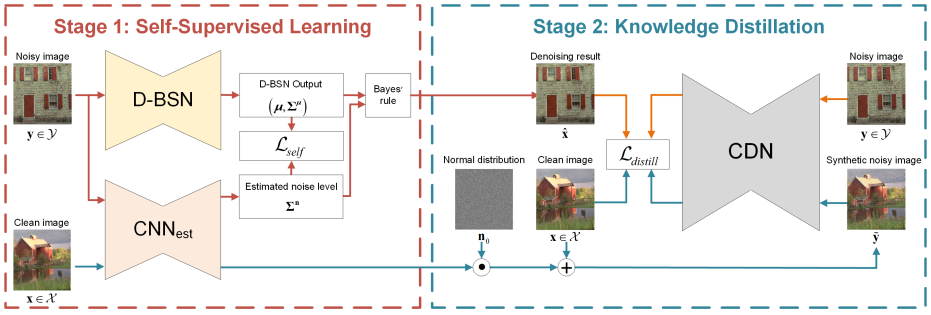


Fig. 2: Illustration of our two-stage training scheme involving self-supervised learning and knowledge distillation.

3.1 Two-Stage Training and Knowledge Distillation

This work tackles the task of learning CNN denoisers from an unpaired set of clean and noisy images. Thus, the training set can be given by two independent sets of clean images $\mathcal{X}$ and noisy images $\mathcal{Y}$. Here, $\mathbf{x}$ denotes a clean image from $\mathcal{X}$, and $\mathbf{y}$ a noisy image from $\mathcal{Y}$. With the unpaired setting, both the real noisy observation of $\mathbf{x}$ and the noise-free image of $\mathbf{y}$ are unavailable. Denote by $\tilde{\mathbf{x}}$ the underlying clean image of $\mathbf{y}$. The real noisy image $\mathbf{y}$ can be written as,

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

where $\mathbf{n}$ denotes the noise in $\mathbf{y}$. Following [22], we assume that the image $\mathbf{x}$ is spatially correlated and the noise $\mathbf{n}$ is pixel-independent and signal-dependent Gaussian. That is, the noise variance (or noise level) at pixel i is determined only by the underlying noise-free pixel value $\tilde{x}_i$ at pixel i,

$$\mathrm{var}(n_i) = g_{\tilde{\mathbf{x}}}(\tilde{x}_i). \tag{2}$$

Thus, $g_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$ can be regarded as a kind of noise level function (NLF) in multivariate heteroscedastic Gaussian model [11,33]. Instead of linear NLF in [11,33], $g_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$ can be any nonlinear function, and thus is more expressive in noise modeling. We also note that the NLF may vary with images (e.g., AWGN with different variance), and thus image-specific $g_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$ is adopted for the flexibility issue.

With the above problem setting and assumptions, Fig. 2 illustrates our two-stage training scheme involving self-supervised learning and knowledge distillation. In the first stage, we elaborate a novel blind-spot network, i.e., D-BSN, and an image-specific noise model $\mathrm{CNN}_{\mathrm{est}}$ (Refer to Sec. 3.2 for details). Then, self-supervised loss is introduced to jointly train D-BSN and $\mathrm{CNN}_{\mathrm{est}}$ solely based on $\mathcal{Y}$ via maximizing the constrained log-likelihood (Refer to Sec. 3.3 for details). For a given real noisy image $\mathbf{y} \in \mathcal{Y}$, D-BSN and $\mathrm{CNN}_{\mathrm{est}}$ collaborate to produce the first stage denoising result $\hat{\mathbf{x}}_{\mathbf{y}}$ and the estimated NLF $g_{\mathbf{y}}(\mathbf{y})$. It is worth noting that we modify the NLF in Eqn. (2) by defining it on the noisy image $\mathbf{y}$ for practical feasibility.

In the second stage, we adopt the knowledge distillation strategy, and exploit $\mathcal{X}$, $\mathcal{Y}$, $\hat{\mathcal{X}}^{(1)} = \{\hat{\mathbf{x}}_{\mathbf{y}} | \mathbf{y} \in \mathcal{Y}\}$, and the set of image-specific NLFs $\{g_{\mathbf{y}}(\mathbf{y}) | \mathbf{y} \in \mathcal{Y}\}$ to distill a state-of-the-art deep denoising network in a fully-supervised manner. On

the one hand, for a given clean image $\mathbf{x} \in \mathcal{X}$, we randomly select an image-specific NLF $g_{\mathbf{y}}(\mathbf{y})$, and use $g_{\mathbf{y}}(\mathbf{x})$ to generate a NLF for $\mathbf{x}$. Denote by $\mathbf{n}_0 \sim \mathcal{N}(0, 1)$ a random Gaussian noise of zero mean and one variance. The synthetic noisy image $\tilde{\mathbf{y}}$ corresponding to $\mathbf{x}$ can then be obtained by,

$$\tilde{\mathbf{y}} = \mathbf{x} + g_{\mathbf{y}}(\mathbf{x}) \cdot \mathbf{n}_0. \tag{3}$$

Consequently, we build the first set of paired noisy-clean images $\{(\mathbf{x}, \tilde{\mathbf{y}})|\mathbf{x} \in \mathcal{X}\}$. On the other hand, given a real noisy image $\mathbf{y}$, we have its denoising result $\hat{\mathbf{x}}_{\mathbf{y}}$ in the first stage, thereby forming the second set of paired noisy-clean images $\{(\hat{\mathbf{x}}_{\mathbf{y}}, \mathbf{y})|\mathbf{y} \in \mathcal{Y}\}$.

The above two paired sets are then used to distill a state-of-the-art convolutional denoising network (CDN) by minimizing the following loss,

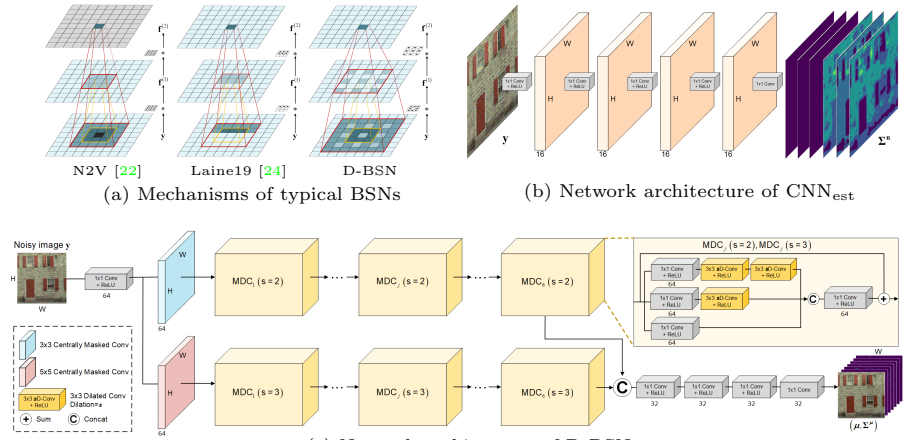
$$\mathcal{L}_{distill} = \sum\nolimits_{\mathbf{x}\in\mathcal{X}} \|\mathrm{CDN}(\tilde{\mathbf{y}}) - \mathbf{x}\|^2 + \lambda \sum\nolimits_{\mathbf{y}\in\mathcal{Y}} \|\mathrm{CDN}(\mathbf{y}) - \hat{\mathbf{x}}_{\mathbf{y}}\|^2, \tag{4}$$

where $\lambda = 0.1$ is the tradeoff parameter. We note that the two paired sets are complementary, and both benefit the denoising performance. In particular, for $\{(\mathbf{x}, \tilde{\mathbf{y}})|\mathbf{x} \in \mathcal{X}\}$, the synthetic noisy image $\tilde{\mathbf{y}}$ may not fully capture real noise complexity when the estimated image-specific noise model $\mathrm{CNN}_{\mathrm{est}}$ is not accurate. Nonetheless, the clean image $\mathbf{x}$ are real, which is beneficial to learn denoising network with visually pleasing result and fine details. As for $\{(\hat{\mathbf{x}}_{\mathbf{y}}, \mathbf{y})|\mathbf{y} \in \mathcal{Y}\}$, the noisy image $\mathbf{y}$ is real, which is helpful in compensating the estimation error in noise model. The denoising result $\hat{\mathbf{x}}_{\mathbf{y}}$ in the first stage may suffer from the over-smoothing effect, which, fortunately, can be mitigated by the real clean images in $\{(\mathbf{x}, \tilde{\mathbf{y}})|\mathbf{x} \in \mathcal{X}\}$. In our two-stage training scheme, the convolutional denoising network CDN can be any existing CNN denoisers, and we consider MWCNN [27] as an example in our implementation.

Our two-stage training scheme offers a novel and effective approach to train CNN denoisers with unpaired learning. In contrast to GAN-based method [8], we present a self-supervised method for joint estimation of denoising result and image-specific noise model. Furthermore, knowledge distillation with two complementary paired sets is exploited to learn a deep denoising network in a fully-supervised manner. The network structures and loss function of our self-supervised learning method are also different with [22–24], which will be explained in the subsequent subsections.

3.2 D-BSN and $\mathrm{CNN}_{\mathrm{est}}$ for Self-Supervised Learning

Blind-spot network (BSN) generally is required for self-supervised learning of CNN denoisers. Among existing BSN solutions, N2V [22] is computationally very inefficient in training. Laine et al. [24] greatly multigate the efficiency issue, but still require four network branches or four rotated versions of each input image, thereby leaving some leeway to further improve training efficiency. To tackle this issue, we elaborately incorporate dilated convolution with FCN to design a D-BSN. Besides, Laine et al. [24] assume the form of noise distribution is known, e.g., AWGN and Poisson noise, and adopt a U-Net to estimate the parameters of noise model. In comparison, our model is based on a more general



(a) Mechanisms of typical BSNs

(b) Network architecture of CNN_{est}

(c) Network architecture of D-BSN

Fig. 3: Mechanisms of BSNs, and network structures of D-BSN and CNN_{est}.

assumption that the noise $\mathbf{n}$ is pixel-independent, signal-dependent, and image-specific. To meet this assumption, we exploit a FCN with 1×1 convolution (i.e., CNN_{est}) to produce an image-specific NLF for noise modeling. In the following, we respectively introduce the structures of D-BSN and CNN_{est}.

D-BSN. For the output at a position, the core of BSN is to exclude the effect of the input value at the same position (i.e., blind-spot requirement). For the first convolution layer, we can easily meet this requirement using masked convolution [32]. Denote by $\mathbf{y}$ a real noisy image, and $\mathbf{w}_k$ the k-th 3×3 convolutional kernel. We introduce a 3×3 binary mask $\mathbf{m}$, and assign 0 to the central element of $\mathbf{m}$ and 1 to the others. The centrally masked convolution is then defined as,

$$\mathbf{f}_k^{(1)} = \mathbf{y} * (\mathbf{w}_k \circ \mathbf{m}) \,, \tag{5}$$

where $\mathbf{f}_k^{(1)}$ denotes the k-th channel of feature map in the first layer, $*$ and $\circ$ denotes the convolution operator and element-wise product, respectively. Obviously, the blind-spot requirement can be satisfied for $\mathbf{f}_k^{(1)}$, but will certainly be broken when further stacking centrally masked convolution layers.

Fortunately, as shown in Fig. 3(a), the blind-spot requirement can be maintained by stacking dilated convolution layers with scale factor $s = 2$ upon 3×3 centrally masked convolution. Analogously, it is also feasible to stacking dilated convolution layers with $s = 3$ upon 5×5 centrally masked convolution. Moreover, 1×1 convolution and skip connection also do not break the blind-spot requirement. Thus, we can leverage centrally masked convolution, dilated convolution, and 1×1 convolution to elaborate a FCN, i.e., D-BSN, while satisfying the blind-spot requirement. In comparison to [24], neither four network branches or four rotated versions of input image are required by our D-BSN. Detailed description for the blind-spot mechanisms illustrated in Fig.3(a) can be found in the supplementary materials.

Fig. 3(c) illustrates the network structure of our D-BSN. In general, our D-BSN begins with a 1×1 convolution layer, and follows by two network branches.

Each branch is composed of a 3×3 (5×5) centrally masked convolution layer following by six multiple dilated convolution (MDC) modules with $s = 2$ ($s = 3$). Then, the feature maps of the two branches are concatenated and three 1×1 convolution layers are further deployed to produce the network output. As shown in Fig. 3(c), the MDC module adopts a residual learning formulation and involves three sub-branches. In these sub-branches, zero, one, and two 3×3 convolution layers are respectively stacked upon a 1×1 convolution layer. The outputs of these sub-branches are then concatenated, followed by another 1×1 convolution layer, and added with the input of the MDC module. Then, the last 1×1 convolution layer is deployed to produce the output feature map. Finally, by concatenating the feature maps from the two network branches, we further apply three 1×1 convolution layers to produce the D-BSN output. Please refer to Fig. 3(c) for the detailed structure of D-BSN.

$\mathrm{CNN}_{\mathrm{est}}$. The noise is assumed to be conditionally pixel-wise independent given the underlying clean image. We assume that the noise is signal-dependent multivariate Gaussian with the NLF $g_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$, and further require that the NLF is image-specific to improve the model flexibility. Taking these requirements into account, we adopt a FCN architecture $\mathrm{CNN}_{\mathrm{est}}$ with 1×1 convolution to learn the noise model. Benefited from all 1×1 convolution layers, the noise level at a position can be guaranteed to only depends on the input value at the same position. Note that the input of $g_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$ is a clean image and we only have noisy images in self-supervised learning. Thus, $\mathrm{CNN}_{\mathrm{est}}$ takes the noisy image as the input and learns the NLF $g_{\mathbf{y}}(\mathbf{y})$ to approximate $g_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$. For an input image of C channels ($C = 1$ for gray level image and 3 for color image), the output at a position i is a $C \times C$ covariance matrix $\mathbf{\Sigma}_i^{\mathbf{n}}$, thereby making it feasible in modeling channel-correlated noise. Furthermore, we require each noisy image has its own network parameters in $\mathrm{CNN}_{\mathrm{est}}$ to learn image specific NLF. From Fig. 3(b), our $\mathrm{CNN}_{\mathrm{est}}$ consists of five 1×1 convolution layers of 16 channels. And the ReLU nonlinearity [21] is deployed for all convolution layers except the last one.

3.3 Self-Supervised Loss and Bayes Denoising

In our unpaired learning setting, the underlying clean image $\tilde{\mathbf{x}}$ and ground-truth NLF of real noisy image $\mathbf{y}$ are unavailable. We thus resort to self-supervised learning to train D-BSN and $\mathrm{CNN}_{\mathrm{est}}$. For a given position i, we have $y_i = \tilde{x}_i + n_i$ with $n_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i^{\mathbf{n}})$. Here, y_i, $\tilde{x}_i$, n_i, and $\mathbf{0}$ all are $C \times 1$ vectors. Let $\boldsymbol{\mu}$ be the directly predicted clean image by D-BSN. We assume $\boldsymbol{\mu} = \tilde{\mathbf{x}} + \mathbf{n}^{\boldsymbol{\mu}}$ with $n_i^{\boldsymbol{\mu}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i^{\boldsymbol{\mu}})$, and further assume that n_i and μ_i are independent. It is noted that $\boldsymbol{\mu}$ is closer to $\tilde{\mathbf{x}}$ than $\mathbf{y}$, and usually we have $|\mathbf{\Sigma}_i^{\mathbf{n}}| \gg |\mathbf{\Sigma}_i^{\boldsymbol{\mu}}| \approx 0$. Considering that $\tilde{x}_i$ is not available, a new variable $\epsilon_i = y_i - \mu_i$ is introduced and it has $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i^{\mathbf{n}} + \mathbf{\Sigma}_i^{\boldsymbol{\mu}})$. The negative log-likelihood of $y_i - \mu_i$ can be written as,

$$\mathcal{L}_{\epsilon_i} = \frac{1}{2}(y_i - \mu_i)^{\top}(\mathbf{\Sigma}_i^{\mathbf{n}} + \mathbf{\Sigma}_i^{\boldsymbol{\mu}})^{-1}(y_i - \mu_i) + \frac{1}{2}\log|\mathbf{\Sigma}_i^{\mathbf{n}} + \mathbf{\Sigma}_i^{\boldsymbol{\mu}}|, \tag{6}$$

where $|\cdot|$ denotes the determinant of a matrix.

However, the above loss ignores the constraint $|\mathbf{\Sigma}_i^{\mathbf{n}}| \gg |\mathbf{\Sigma}_i^{\boldsymbol{\mu}}| \approx 0$. Actually, when taking this constraint into account, the term $\log|\mathbf{\Sigma}_i^{\mathbf{n}} + \mathbf{\Sigma}_i^{\boldsymbol{\mu}}|$ in Eqn. (6) can be well approximated by its first-order Taylor expansion at the point $\mathbf{\Sigma}_i^{\mathbf{n}}$,

$$\log|\mathbf{\Sigma}_i^{\mathbf{n}} + \mathbf{\Sigma}_i^{\boldsymbol{\mu}}| \approx \log|\mathbf{\Sigma}_i^{\mathbf{n}}| + \operatorname{tr}\left((\mathbf{\Sigma}_i^{\mathbf{n}})^{-1}\mathbf{\Sigma}_i^{\boldsymbol{\mu}}\right), \tag{7}$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. Note that $\mathbf{\Sigma}_i^{\mathbf{n}}$ and $\mathbf{\Sigma}_i^{\boldsymbol{\mu}}$ are treated equally in the left term. While in the right term, smaller $\mathbf{\Sigma}_i^{\boldsymbol{\mu}}$ and larger $\mathbf{\Sigma}_i^{\mathbf{n}}$ are favored based on $\operatorname{tr}\left((\mathbf{\Sigma}_i^{\mathbf{n}})^{-1}\mathbf{\Sigma}_i^{\boldsymbol{\mu}}\right)$, which is consistent with $|\mathbf{\Sigma}_i^{\mathbf{n}}| \gg |\mathbf{\Sigma}_i^{\boldsymbol{\mu}}| \approx 0$.

Actually, μ_i and $\mathbf{\Sigma}_i^{\boldsymbol{\mu}}$ can be estimated as the output of D-BSN at position i, i.e., $\hat{\mu}_i = (\text{D-BSN}_{\mu}(\mathbf{y}))_i$ and $\hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}} = (\text{D-BSN}_{\mathbf{\Sigma}}(\mathbf{y}))_i$. $\mathbf{\Sigma}_i^{\mathbf{n}}$ can be estimated as the output of CNN_{est} at position i, i.e., $\hat{\mathbf{\Sigma}}_i^{\mathbf{n}} = (\text{CNN}_{\text{est}}(\mathbf{y}))_i$. By substituting Eqn. (7) into Eqn. (6), and replacing μ_i, $\mathbf{\Sigma}_i^{\boldsymbol{\mu}}$ and $\mathbf{\Sigma}_i^{\mathbf{n}}$ with the network outputs, we adopt the constrained negative log-likelihood for learning D-BSN and CNN_{est},

$$\mathcal{L}_{self} = \sum\nolimits_i \frac{1}{2}\left\{(y_i - \hat{\mu}_i)^{\top}(\hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}} + \hat{\mathbf{\Sigma}}_i^{\mathbf{n}})^{-1}(y_i - \hat{\mu}_i) + \log|\hat{\mathbf{\Sigma}}_i^{\mathbf{n}}| + \operatorname{tr}\left((\hat{\mathbf{\Sigma}}_i^{\mathbf{n}})^{-1}\hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}}\right)\right\}. \tag{8}$$

After self-supervised learning, given the output $\text{D-BSN}_{\mu}(\mathbf{y})$, $\text{D-BSN}_{\mathbf{\Sigma}}(\mathbf{y})$ and $\text{CNN}_{\text{est}}(\mathbf{y})$, the denoising result in the first stage can be obtained using the Bayes' rule to each pixel,

$$\hat{x}_i = (\hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}} + \hat{\mathbf{\Sigma}}_i^{\mathbf{n}})^{-1}(\hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}} y_i + \hat{\mathbf{\Sigma}}_i^{\mathbf{n}} \hat{\mu}_i). \tag{9}$$

3.4 Extension to Real-world Noisy Photographs

Due to the effect of demosaicking, the noise in real-world photographs is spatially correlated and violates the pixel-independent noise assumption, thereby restricting the direct application of our method. Nonetheless, such assumption is critical in separating signal (spatially correlated) and noise (pixel-independent). Fortunately, the noise is only correlated within a short range. Thus, we can break this dilemma by training D-BSN on the pixel-shuffle downsampled images with factor 4. Considering the noise distributions on sum-images are different, we assign the 16 sub-images to 4 groups according to the Bayer pattern. The results of 16 sub-images are then pixel-shuffle upsampled to form an image of the original size, and the guided filter [16] with radius of 1 and penalty value of 0.01 is applied to obtain the final denoising image. We note that denoising on pixel-shuffle sub-images slightly degrades the performance. Nonetheless, our method can still obtain visually satisfying results on real-world noisy photographs.

4 Experimental Results

In this section, we first describe the implementation details and conduct ablation study of our method. Then, extensive experiments are carried out to evaluate our method on synthetic and real-world noisy images. The evaluation is performed on a PC with Intel(R) Core (TM) i9-7940X CPU @ 3.1GHz and an Nvidia Titan RTX GPU. The source code and pre-trained models will be publicly available.

Table 1: Average PNSR(dB) results of different methods on the BSD68 dataset with noise levels 15, 25 and 50, and heteroscedastic Gaussian (HG) noise with $\alpha = 40$, $\delta = 10$.

Noise	Para.	BM3D [9]	DnCNN [40]	NLRN [26]	N3Net [34]	N2V [22]	Laine19 [24]	GCBD [8]	D-BSN (ours)	N2C [27]	N2N [25]	Ours (full)
AWGN	$\sigma=15$	31.07	31.72	31.88	-	-	-	31.37	31.24	31.86	31.71	31.82
	$\sigma=25$	28.57	29.23	29.41	29.30	27.71	29.27	28.83	28.88	29.41	29.33	29.38
	$\sigma=50$	25.62	26.23	26.47	26.39	-	-	-	25.81	26.53	26.52	26.51
HG	$\alpha=40$ $\delta=10$	23.84	-	-	-	-	-	-	29.13	30.16	29.53	30.10

4.1 Implementation details

Our unpaired learning consists of two stages: (i) self-supervised training of D-BSN and CNN_{est} and (ii) knowledge distillation for training MWCNN [27], which are respectively described as follows.

Self-Supervised Training. For synthetic noises, the clean images are from the validation set of ILSVRC2012 (ImageNet) [10] while excluding the images smaller than 256×256. Several basic noise models, e.g., AWGN, multivariate Gaussian, and heteroscedastic Gaussian, are adopted to synthesize the noisy images $\mathcal{Y}$. While for real noisy images, we simply use the testing dataset as $\mathcal{Y}$. During the training, we randomly crop $48,000$ patches with size 80×80 in each epoch and finish the training after 180 epochs. The Adam optimizer is employed to train D-BSN and CNN_{est}. The learning rate is initialized as 1×10^{-4}, and is decayed by factor 10 after every 30 epochs until reaching 1×10^{-7}.

Knowledge Distillation. For gray level images, we adopt the BSD [36] training set as clean image set $\mathcal{X}$. For color images, DIV2K [3], WED [29] and CBSD [36] training set are used to form $\mathcal{X}$. Then, we exploit both $\{(\hat{\mathbf{x}}_{\mathbf{y}}, \mathbf{y}) | \mathbf{y} \in \mathcal{Y}\}$ and $\{(\mathbf{x}, \tilde{\mathbf{y}}) | \mathbf{x} \in \mathcal{X}\}$ to train a state-of-the-art CNN denoiser from scratch. And MWCNN with original setting [27] on learning algorithm is adopted to train the CNN denoiser on our data.

4.2 Comparison of Different Supervision Settings

CNN denoisers can be trained with different supervision settings, such as N2C, N2N [25], N2V [22], Laine19 [24], GCBD [8], our D-BSN and MWCNN(unpaired). For a fair comparison, we retrain two MWCNN models with the N2C and N2N [25] settings, respectively. The results of N2V [22], Laine19 [24] and GCBD [8] are from the original papers.

Results on Gray Level Images. We consider two basic noise models, i.e., AWGN with $\sigma = 15, 25$ and 50, and heteroscedastic Gaussian (HG) [33] $n_i \sim \mathcal{N}(0, \alpha^2 x_i + \delta^2)$ with $\alpha = 40$ and $\delta = 10$. From Table 1, on BSD68 [36] our D-BSN performs better than N2V [22] and on par with GCBD [8], but is inferior to Laine19 [24]. Laine19 [24] learns the denoisers solely from noisy images and does not exploit unpaired clean images in training, making it still poorer than MWCNN(N2C). By exploiting both noisy and clean images, our MWCNN(unpaired)

Table 2: Average PNSR(dB) results of different methods on the CBSD68 dataset with noise levels 15, 25 and 50, heteroscedastic Gaussian (HG) noise with $\alpha = 40$, $\delta = 10$, and multivariate Gaussian (MG) noise.

Noise	Para.	CBM3D [9]	CDnCNN [40]	FFDNet [42]	CBDNet [15]	Laine19 [24]	D-BSN (ours)	N2C [27]	N2N [25]	Ours (full)
AWGN	$\sigma = 15$	33.52	33.89	33.87	-	-	33.56	34.08	33.76	34.02
	$\sigma = 25$	30.71	30.71	31.21	-	31.35	30.61	31.40	31.22	31.40
	$\sigma = 50$	27.38	27.92	27.96	-	-	27.41	28.26	27.79	28.25
HG	$\alpha = 40$ $\delta = 10$	23.21	-	28.67	30.89	-	30.53	32.10	31.13	31.72
MG	$\mathbf{\Sigma}=75^2\cdot\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ $\lambda_c\in(0,1)$ $\mathbf{U}^T\mathbf{U}=\mathbf{I}$	24.07	-	26.78	21.50	-	26.48	26.89	26.59	26.81

Table 3: Average PNSR(dB) results of different methods on KODAK24 and McMaster datasets.

Dataset	Para.	CBM3D [9]	CDnCNN [40]	FFDNet [42]	Laine19 [24]	D-BSN (ours)	Ours (full)
KODAK24	$\sigma = 15$	34.28	34.48	34.63	-	33.74	34.82
	$\sigma = 25$	31.68	32.03	32.13	32.33	31.64	32.35
	$\sigma = 50$	28.46	28.85	28.98	-	28.69	29.36
McMaster	$\sigma = 15$	34.06	33.44	34.66	-	33.85	34.87
	$\sigma = 25$	31.66	31.51	32.35	32.52	31.56	32.54
	$\sigma = 50$	28.51	28.61	29.18	-	28.87	29.58

(also Ours(full) in Table 1) outperforms both Laine19 [24] and MWCNN(N2N) in most cases, and is on par with MWCNN(N2C). In terms of training time, Laine19 takes about 14 hours using four Tesla V100 GPUs on NVIDIA DGX-1 servers. In contrast, our D-BSN takes about 10 hours on two 2080Ti GPUs and thus is more efficient. In terms of testing time, Our MWCNN(unpaired) takes 0.020s to process a 320×480 image, while N2V [22] needs 0.034s and Laine19 [24] spends 0.044s.

Results on Color Images. Besides AWGN and HG, we further consider another noise model, i.e., multivariate Gaussian (MG) [42] $n \sim \mathcal{N}(0, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = 75^2 \cdot \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. Here, $\mathbf{U}$ is a random unitary matrix, $\mathbf{\Lambda}$ is a diagonal matrix of three random values in the range $(0, 1)$. GCBD [8] and N2V [22] did not report their results on color image denoising. On CBSD68 [36] our MWCNN(unpaired) is consistently better than Laine19 [24], and notably outperforms MWCNN(N2N) [25] for HG noise. We have noted that both MWCNN(unpaired) and D-BSN perform well in handling multivariate Gaussian with cross-channel correlation.

4.3 Experiments on Synthetic Noisy Images

In this subsection, we assess our MWCNN(unpaired) in handling different types of synthetic noise, and compare it with the state-of-the-art image denoising methods. The competing methods include BM3D [9], DnCNN [40], NLRN [26] and N3Net [34] for gray level image denoising on BSD68 [36], and CBM3D [9], CDnCNN [40], FFDNet [42] and CBDNet [15] for color image denoising on BSD68 [36], KODAK24 [12], McMaster [43].

Table 4: The quantitative results (PSNR/SSIM) of different methods on the CC15 and DND (sRGB images) datasets.

Method	BM3D [9]	DnCNN [40]	CBDNet [15]	VDN [39]	N2S [5]	N2V [22]	GCBD[8]	MWCNN(unpaired)
Supervised	-	Yes	Yes	Yes	Not	Not	Not	Not
CC15	35.19 0.9063	33.86 0.8636	36.47 0.9392	- -	35.38 0.9204	35.27 0.9158	- -	35.90 0.9370
DND	34.51 0.8507	32.43 0.7900	38.05 0.9421	39.38 0.9518	- -	- -	35.58 0.9217	37.93 0.9373

AWGN. From Table 1, it can be seen that our MWCNN(unpaired) performs on par with NLRN [26] and better than BM3D [9], DnCNN [40] and N3Net [34]. Tables 2 and 3 list the results of color image denoising. Benefited from unpaired learning and the use of MWCNN [27], our MWCNN(unpaired) outperforms CBM3D [9], CDnCNN [40] and FFDNet [42] by a non-trivial margin for all noise levels and on the three datasets.

Heteroscedastic Gaussian. We further test our MWCNN(unpaired) in handling heteroscedastic Gaussian (HG) noise which is usually adopted for modeling RAW image noise. It can be seen from Table 2 that all the CNN denoisers outperform CBM3D [9] by a large margin on CBSD68. CBDNet [15] takes both HG noise and in-camera signal processing pipeline into account when training the deep denoising network, and thus is superior to FFDNet [42]. Our MWCNN(unpaired) can leverage the progress in CNN denoisers and well exploit the unpaired noisy and clean images, and achieves a PSNR gain of 0.8dB against CBDNet [15].

Multivariate Gaussian. Finally, we evaluate our MWCNN(unpaired) in handling multivariate Gaussian (MG) noise with cross-channel correlation. Some image processing operations, e.g., image demosaicking, may introduce cross-channel correlated noise. From Table 2, FFDNet [42] is flexible in handle HG noise, while our unpaired learning method, MWCNN(unpaired), performs on par with FFDNet [42] on CBSD68.

4.4 Experiments on Real-world Noisy Photographs

Finally, we conduct comparison experiments on two widely adopted datasets of real-world noisy photographs, i.e, DND [33] and CC15 [31]. Our methods is compared with both traditional denoising method, i.e., CBM3D [9], deep Gaussian denoiser, i.e., DnCNN [40], deep blind denoisers, i.e., CBDNet [15] and VDN [39], and unsupervised learning methods, i.e., GCBD [8], N2S [25] and N2V [22], in terms of both quantitative and qualitative results. The average PSNR and SSIM metrics are presented in Table 4. On CC15, our MWCNN(unpaired) outperforms the other unsupervised learning methods (i.e., N2S [25] and N2V [22]) by a large margin (0.5dB in PSNR). On DND, our MWCNN(unpaired) achieves a PSNR gain of 2.3dB against GCBD. The results clearly show the merits of our methods in exploiting unpaired noisy and clean images. Actually, our method

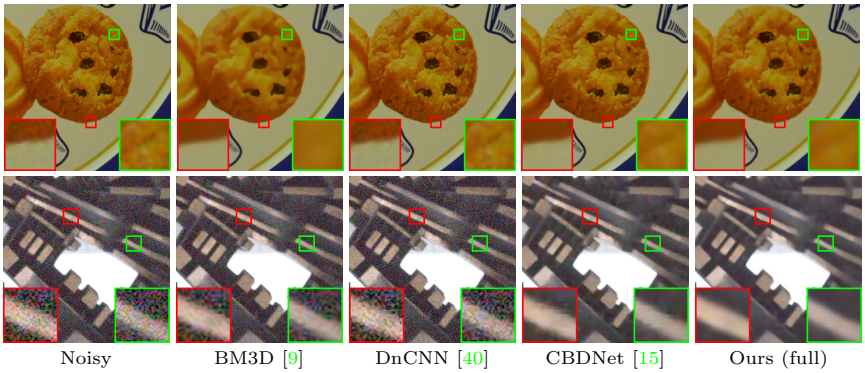


Fig. 4: Denoising results of different methods on real-world images from C-C15(up) and DND(dowm) datasets.

is only inferior to deep models specified for real-world noisy photography, e.g., CBDNet [15] and VDN [39]. Such results should not be criticized considering that our MWCNN(unpaired) has no access to neither the details of ISP [15] and the paired noisy-clean images [39]. Moreover, we adopt the pixel-shuffle down-sampling to decouple spatially correlated noise, which also gives rise to moderate performance degradation of our method. Nonetheless, it can be seen from Fig. 4 that our MWCNN(unpaired) achieves comparable or better denoising results in comparison to all the competing methods on DND and CC15.

5 Concluding Remarks

This paper presented a novel unpaired learning method by incorporating self-supervised learning and knowledge distillation for training CNN denoisers. In self-supervised learning, we proposed a dilated blind-spot network (D-BSN) and a FCN with 1×1 convolution, which can be efficiently trained via maximizing constrained log-likelihood from unorganized collections of noisy images. For knowledge distillation, the estimated denoising image and noise model are used to distill the state-of-the-art CNN denoisers, such as DnCNN and MWCNN. Experimental results showed that the proposed method is effective on both images with synthetic noise (e.g., AWGN, heteroscedastic Gaussian, multivariate Gaussian) and real-world noisy photographs.

Compared with [22, 24] and GAN-based unpaired learning of CNN denoisers [8], our method has several merits in efficiently training blind-spot networks and exploiting unpaired noisy and clean images. However, there remain a number of challenges to be addressed: (i) our method is based on the assumption of pixel-independent heteroscedastic Gaussian noise, while real noise can be more complex and spatially correlated. (ii) In self-supervised learning, estimation error may be unavoidable for clean images and noise models, and it is interesting to develop robust and accurate distillation of CNN denoisers.

Acknowledgement. This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No.s 61671182, U19A2073.

References

1. Abdelhamed, A., Brubaker, M.A., Brown, M.S.: Noise Flow: Noise modeling with conditional normalizing flows. In: ICCV. pp. 3165–3173 (2019) 5
2. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: CVPR. pp. 1692–1700 (2018) 2, 4
3. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPR Workshops (2017) 11
4. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: ICCV. pp. 3155–3164 (2019) 4
5. Batson, J., Royer, L.: Noise2self: Blind denoising by self-supervision pp. 524–533 (2019) 5, 13
6. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: CVPR. pp. 11036–11045 (2019) 2, 4
7. Bulat, A., Yang, J., Tzimiropoulos, G.: To learn image super-resolution, use a gan to learn how to do image degradation first. In: ECCV. pp. 185–200 (2018) 3
8. Chen, J., Chen, J., Chao, H., Yang, M.: Image blind denoising with generative adversarial network based noise modeling. In: CVPR. pp. 3155–3164 (2018) 2, 3, 4, 5, 7, 11, 12, 13, 14
9. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. TIP **16**(8), 2080–2095 (2007) 2, 11, 12, 13, 14
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009) 11
11. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K.: Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. TIP **17**(10), 1737–1754 (2008) 4, 6
12. Franzen, R.: Kodak lossless true color image suite. source: http://r0k. us/graphics/kodak **4** (1999) 12
13. Grossberg, M.D., Nayar, S.K.: Modeling the space of camera response functions. TPAMI **26**(10), 1272–1282 (2004) 4
14. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: CVPR. pp. 2862–2869 (2014) 2
15. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: CVPR. pp. 1712–1722 (2019) 2, 4, 12, 13, 14
16. He, K., Sun, J., Tang, X.: Guided image filtering. In: ECCV. pp. 1–14 (2010) 10
17. Ian J. Goodfellow, Jean Pouget-Abadie, M.M.B.X.D.W.F.S.O.A.C., Bengio, Y.: Generative Adversarial Networks. In: NIPS. p. 2672C2680 (2014) 3
18. Islam, M.T., Rahman, S.M., Ahmad, M.O., Swamy, M.: Mixed gaussian-impulse noise reduction from images using convolutional neural network. Signal Processing: Image Communication **68**, 26–41 (2018) 2
19. Jia, X., Liu, S., Feng, X., Zhang, L.: FOCNet: A fractional optimal control network for image denoising. In: CVPR. pp. 6054–6063 (2019) 4
20. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: NIPS. pp. 10215–10224 (2018) 4
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012) 9
22. Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: CVPR. pp. 2129–2137 (2019) 2, 4, 5, 6, 7, 8, 11, 12, 13, 14

23. Krull, A., Vicar, T., Jug, F.: Probabilistic noise2void: Unsupervised content-aware denoising. arXiv preprint arXiv:1906.00651 (2019) 2, 5, 7
24. Laine, S., Karras, T., Lehtinen, J., Aila, T.: High-quality self-supervised deep image denoising. In: NIPS. pp. 6968–6978 (2019) 2, 3, 4, 5, 7, 8, 11, 12, 14
25. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. In: ICML. pp. 2965–2974 (2018) 2, 5, 11, 12, 13
26. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: NIPS. pp. 1673–1682 (2018) 1, 3, 4, 11, 12, 13
27. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-CNN for image restoration. In: CVPR Workshops. pp. 773–782 (2018) 1, 2, 3, 4, 7, 11, 12, 13
28. Liu, X., Tanaka, M., Okutomi, M.: Practical signal-dependent noise parameter estimation from a single noisy image. TIP **23**(10), 4361–4371 (2014) 4
29. Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo exploration database: New challenges for image quality assessment models. TIP **26**(2), 1004–1016 (2016) 11
30. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NIPS. pp. 2802–2810 (2016) 1, 2, 3, 4
31. Nam, S., Hwang, Y., Matsushita, Y., Joo Kim, S.: A holistic approach to cross-channel image noise modeling and its application to image denoising. In: CVPR. pp. 1683–1691 (2016) 2, 4, 13
32. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel Recurrent Neural Networks. In: ICML. p. 1747C1756 (2016) 8
33. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: CVPR. pp. 1586–1595 (2017) 2, 3, 4, 6, 11, 13
34. Plötz, T., Roth, S.: Neural Nearest Neighbors Networks. In: NIPS. pp. 1087–1098 (2018) 1, 3, 4, 11, 12, 13
35. Remez, T., Litany, O., Giryes, R., Bronstein, A.M.: Class-aware fully convolutional gaussian and poisson denoising. TIP **27**(11), 5707–5722 (2018) 2
36. Roth, S., Black, M.J.: Fields of experts. IJCV **82**(2), 205 (2009) 11, 12
37. Soltanayev, S., Chun, S.Y.: Training deep learning based denoisers without ground truth data. In: NIPS. pp. 3257–3267 (2018) 5
38. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. pp. 3147–3155 (2017) 2, 3, 4
39. Yue, Z., Yong, H., Zhao, Q., Meng, D., Zhang, L.: Variational Denoising Network: Toward blind noise modeling and removal. In: NIPS. pp. 1688–1699 (2019) 4, 13, 14
40. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. TIP **26**(7), 3142–3155 (2017) 1, 2, 3, 4, 11, 12, 13, 14
41. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3929–3938 (2017) 3, 4
42. Zhang, K., Zuo, W., Zhang, L.: FFDNet: Toward a fast and flexible solution for cnn-based image denoising. TIP **27**(9), 4608–4622 (2018) 2, 3, 12, 13
43. Zhang, L., Wu, X., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic imaging **20**(2), 023016 (2011) 12

44. Zhussip, M., Soltanayev, S., Chun, S.Y.: Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior. In: CVPR. pp. 10255–10264 (2019) 5