This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LAN: Learning to Adapt Noise for Image Denoising

Changjin Kim
Dept. of Computer Science
Hanyang University
chjinny@hanyang.ac.kr

Tae Hyun Kim
Dept. of Computer Science
Hanyang University
taehyunkim@hanyang.ac.kr

Sungyong Baik[†]
Dept. of Data Science
Hanyang University
dsybaik@hanyang.ac.kr

Abstract

Removing noise from images, a.k.a image denoising, can be a very challenging task since the type and amount of noise can greatly vary for each image due to many factors including a camera model and capturing environments. While there have been striking improvements in image denoising with the emergence of advanced deep learning architectures and real-world datasets, recent denoising networks struggle to maintain performance on images with noise that has not been seen during training. One typical approach to address the challenge would be to adapt a denoising network to new noise distribution. Instead, in this work, we shift our focus to adapting the input noise itself, rather than adapting a network. Thus, we keep a pretrained network frozen, and adapt an input noise to capture the fine-grained deviations. As such, we propose a new denoising algorithm, dubbed Learning-to-Adapt-Noise (LAN), where a learnable noise offset is directly added to a given noisy image to bring a given input noise closer towards the noise distribution a denoising network is trained to handle. Consequently, the proposed framework exhibits performance improvement on images with unseen noise, displaying the potential of the proposed research direction.

1. Introduction

Noise, an unwanted byproduct during image processing, can cause severe degradation not only in image quality but also in high-level computer vision tasks. As such, noise has been a target to eliminate in the field of image denoising. One of the main challenges in image denoising is how to distinguish a noise from an original source without knowing the noise distribution a priori.

To tackle such a challenging problem, a myriad of learning-based models have been proposed to learn to remove noise that follows known distributions. To do so, models are trained on datasets composed of pairs of clean images and corresponding noisy images that are synthesized by adding noise to clean images. Under such formulation, learning-based models have prominently marked progress in the performance of image denoising, ever since the emergence of convolutional neural networks (CNN) and its advanced architectures designed for image denoising [5, 11, 52, 54].

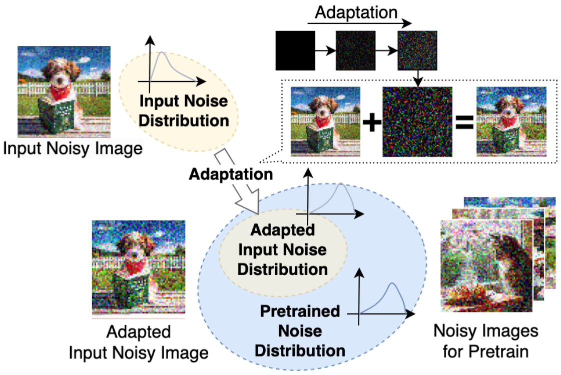

Adaptation
Input Noise Distribution
Input Noisy Image
Adaptation
Adapted Input Noise Distribution
Pretrained Noise Distribution
Adapted Input Noisy Image
Noisy Images for Pretrain

Figure 1. **Overview of the motivation of our framework, Learning-to-Adapt-Noise (LAN)**. Instead of adapting a denoising network to unseen noise, LAN adapts the input noise itself by directly learning to offset the deviations between the unseen noise and the noise distribution a denoising network is trained on.

While early CNN-based models have brought substantial improvements, such impressive results are only limited to images with known noise distribution. Since early neural networks are trained on images with fixed and known noise distributions, models fail to generalize to images that contain unseen noise. Considering that images often contain unseen/unknown noise, due to new environments and camera models, in real-world applications, the incapability of handling such noise is a critical shortcoming that needs to be addressed.

To bridge the gap between the performance on controlled environments and real-world environments, various works have focused on constructing datasets consisting of real-world clean-noisy image pairs [1, 3, 4, 37, 39, 48]. For instance, SIDD [1] dataset, one of widely accepted and used datasets, is constructed by capturing real-world noisy images with various smartphone cameras under different environment (e.g., ISO, shutter speed, and aperture settings).

[†]Corresponding author.

However, not only does the acquisition of such real-world datasets require laborious human effort, but it remains difficult to achieve the generalization to unseen noise that is substantially different from noise distribution within the train data.

In parallel with efforts in collecting real-world noise dataset, there has been several attempts in improving the performance on real-world noise with different approaches, which can be mainly classified into two methodologies: generative modeling and self-supervised learning. Generative modeling approaches employ generative models to synthesize realistic noisy images [2, 9, 20, 23, 50, 51]. However, generative modeling approaches require clean images, from which noisy images are conditionally generated.

On the other hand, self-supervised learning based blind denoising approaches [6, 24, 27, 29, 30, 35, 40, 45] make use of statistical assumptions that allow models to be trained only using noisy images. Such self-supervised learning formulations enable the model to be finetuned and adapted to a given image with unseen noise. While there has been a lot of efforts in designing more practical and effective self-supervision tasks (e.g., loss functions, targets, etc.), the focus has been less on how models are finetuned with given self-supervision tasks. Few works have attempted to adapt the model to a given noisy image [16, 28].

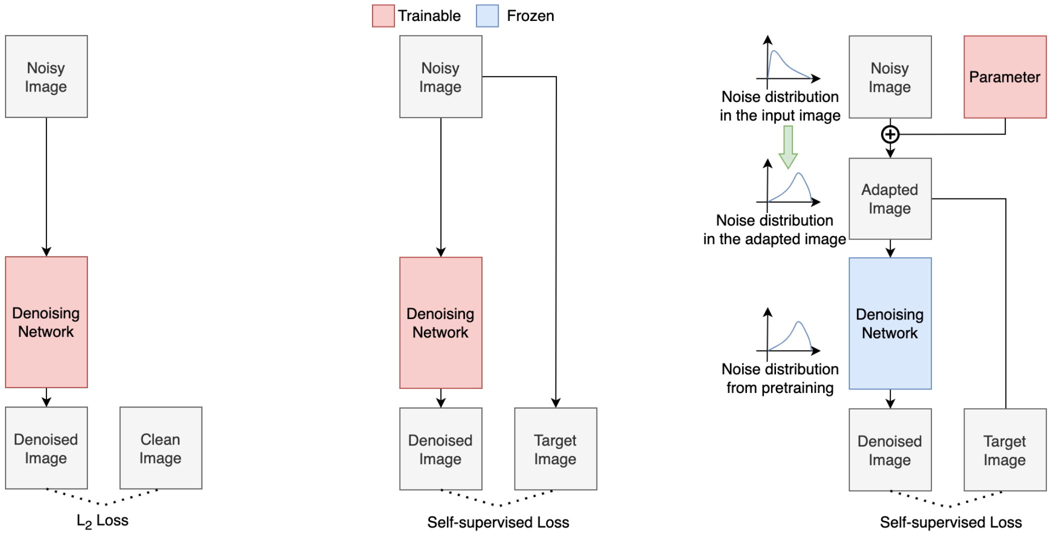
In this work, instead of adapting the model to handle fine-grained deviations in input image that unseen noise brings, we propose to learn to directly offset the deviations between the unseen noise distribution in the input and the noise distribution expected by a pretrained denoising network, as shown in Figure 1. To this end, we propose a new framework, dubbed Learning to Adapt Noise (LAN), which adapts an input noise with a learnable pixel-wise offset that is trained with the aid of self-supervision tasks. Orthogonal and complementary to self-supervised learning algorithms, our framework is shown to bring substantial improvements, in comparison to a model adaptation approach.

2. Related works

Single image denoising. Image denoising is a crucial area of computer vision research. Early approaches include total variation-based denoising [41], sparse coding-based denoising [34], and self-similarity-based denoising methods [7, 12]. With the success of deep learning in computer vision, many deep learning-based denoising methods have been developed, starting with methods that combine sparse coding and MLP [47]. DnCNN [54] focuses on noise by residual structure. FFDNet [55] uses downsampling and non-uniform noise level maps to achieve a faster and more efficient performance. RIDNet [5] has further improved performance especially on real-world noisy datasets, such as SIDD [1], by incorporating a reinforcement attention module that utilizes global information such as feature attention and local skip connection bases. In recent years, vision transformers [13] have brought significant advancements in image restoration. Several works have employed vision transformers to bring futher improvements to denoising [10, 31, 46, 53].

Blind image denoising. While denoising neural networks achieve high performance, they require pairs of clean and noisy images. However, obtaining clean images can be challenging. Several blind denoising approaches [6, 40, 43, 45] have emerged to tackle unseen noise image denoising without the need for clean images to training network. Early strategy in blind denoising is based on an internal image prior [43]. Recently, blind denoising approaches based on self-supervised learning have endeavored to achieve performance levels comparable to supervised learning. Under the assumption that noise in each image is independent and has a zero-mean distribution, Noise2Noise [30] takes different images of a scene that act as input and target images for training a network, respectively. Noise2Self [6] is another self-supervised learning method that masks the noisy image at regular intervals, using the remaining pixels as an input image and masked pixels as a target image. On the other hand, recent works have focused on creating a input-target pair by subsampling from a single noisy image. Notably, Neighbor2Neighbor [17] is uses a random neighbor sub-sampler to generate input-target pairs for training a network. Zero-Shot Noise2Noise [35] is another self-supervised learning approach that generates input-target pairs by applying a filter that computes the mean of diagonal pixels. By test-time adaptation (TTA) [25, 44, 49] using these self-supervised learning approaches, a pretrained denoising network can be either finetuned to handle unseen noise [16, 28]. However, the misalignment between new unseen noise and noise expected by a pretrained denoising network may lead to suboptimal performance even after adaptation. In this work, we approach the problem from a different perspective: adapt a new noisy image itself to reduce the misalignment itself. Figure 2 outlines the major differences between our framework LAN and standard approaches.

Domain adaptive image translation. Adaptation of an input noisy image to handle the misalignment shares motivations with image translation for domain adaptation. Image translation aims to learn to translate images from a source domain to a target domain, through the aid of generative models [15, 19, 22, 26, 32, 33, 56]. While our proposed framework shares some similarities with these works from the perspective of domain adaptation, domain adaptation approaches either require a large amount of target-domain images or require a large amount of images from various domains in order to train a large generative model for image translation. On the other hand, we do not need a separate generative model for image translation. We also do not as-



(a) Pretraining (b) Fine-tuning (c) LAN (Ours)

Figure 2. **Overview of conventional methods and our framework, Learning-to-Adapt-Noise (LAN, ours).** (a) Pretraining of a denoising network is done with pairs of noisy-clean images, with standard L_2 loss. (b) Fine-tuning of a whole denoising network is done with only a given noisy image via self-supervision loss function, such as ZS-N2N [35], to handle unseen noise in the image. (c) Learning-to-Adapt-Noise (LAN, ours) is similar to fine-tuning in that only a give noisy image is used with self-supervision loss. However, our method keeps the whole denoising network frozen and only adapts a given noisy image to handle unseen noise.

sume the availability of a large amount of images with new unseen noise distribution. In fact, we train a learnable offset that is directly added to each input image for adaptation, without the need for an additional network for image translation or the need for other images. Furthermore, we believe it is an interesting perspective to draw connections between image denoising and domain adaptation.

Adversarial attack on image domain. Directly optimizing the offset added to an image exhibit similar characteristics as adversarial attack. Adversarial attack is often performed via adversarial examples generated by adding a learnable noise to the input, where noise is optimized to make the neural network output deviate from its original value [8, 42]. However, while adversarial attacks aim to degrade the model performance, our work focuses on improving the performance. Regardless, we introduce a new interesting perspective, where denoising performance can be improved with learnable anti-adversarial noise.

3. Proposed method

3.1. Preliminaries

Problem formulation. In this work, we aim to tackle a scenario, where a new input image contains noise that follows a different distribution from the distribution a denoising network is trained with. Formally, a denoising network f with parameters $\boldsymbol{\theta}$ is pretrained with pairs of clean images $\boldsymbol{x}^s$ and their noisy counterparts $\boldsymbol{y}^s$ that contain noise $\boldsymbol{e}^s$ assumed to follow a certain distribution $\mathcal{D}^s$, constructed as follows:

$$\boldsymbol{y}^s = \boldsymbol{x}^s + \boldsymbol{e}^s, \qquad \text{where} \quad \boldsymbol{e}^s \sim \mathcal{D}^s. \tag{1}$$

In particular, a denoising network $f_{\boldsymbol{\theta}}$ is trained to map noisy images $\boldsymbol{y}^s$ with certain noise $\boldsymbol{e}^s$ to their clean counterparts $\boldsymbol{x}^s$ via minimizing the empirical loss as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left\| f_{\boldsymbol{\theta}}(\boldsymbol{y}^s) - \boldsymbol{x}^s \right\|_2^2\right]. \tag{2}$$

During test time, we expect that a new input noisy image $\boldsymbol{y}^u$ contains an unknown clean image $\boldsymbol{x}^u$ with unseen noise $\boldsymbol{e}^u$ that follows a distribution $\mathcal{D}^u$ that is different from the distribution seen during training (i.e., $\mathcal{D}^s$):

$$\boldsymbol{y}^u = \boldsymbol{x}^u + \boldsymbol{e}^u, \qquad \text{where} \quad \boldsymbol{e}^u \sim \mathcal{D}^u. \tag{3}$$

Under such formulation, challenges arise due to the domain misalignment between a noise distribution a denoising network is trained on and a new noise distribution encountered during the test phase.

Self-supervised learning. One approach to handle such misalignment would be to adapt a pretrained network to given noisy image. However, only a noisy image $\boldsymbol{y}^u$ is available while a clean image $\boldsymbol{x}^u$ is unavailable, making it difficult to train or adapt a denoising network $f_{\boldsymbol{\theta}}$ via Equation 2.

To extract an underlying clean image from a noisy image, a few recent works have introduced self-supervised learning approaches [6, 18, 30, 35], given the following assumption: clean image pixels and noise pixels exhibit different attributes. Namely, clean image pixels are highly correlated within local regions, whereas the noise pixels are independent. Upon the assumption, two independent noisy images, $\boldsymbol{y}_1^u$ and $\boldsymbol{y}_2^u$, are created out of the same scene from the original noisy image $\boldsymbol{y}^u$ through two different transformations D_1 and D_2, such as downsampling [18, 35]. Then, to approximate the optimization in Equation 2 without clean images, one of two independent noisy images is used as an input to a network while the other as a target:

$$\mathcal{L}_{\text{self}} = \left\| f_{\boldsymbol{\theta}}(D_1(\boldsymbol{y}^u)) - D_2(\boldsymbol{y}^u) \right\|_2^2 . \tag{4}$$

3.2. Learning to adapt noise (LAN)

In this work, we aim to utilize a pretrained denoising network $f_{\boldsymbol{\theta}^*}$ to remove new unseen noise $\boldsymbol{e}^u$ from a given noisy image $\boldsymbol{y}^u$. However, when a denoising network $f_{\boldsymbol{\theta}^*}$ is trained with noise $\boldsymbol{e}^s \sim \mathcal{D}^s$ via Equation 2 while a new noisy image contains noise $\boldsymbol{e}^u \sim \mathcal{D}^u$, there arises misalignment between new input and what a pretrained network expects. Such misalignment may worsen as the difference between seen noise distribution $\mathcal{D}^s$ and unseen noise distribution $\mathcal{D}^u$ is larger. The misalignment may lead to suboptimal performance, when directly finetuning a pretrained denoising network $f_{\boldsymbol{\theta}^*}$ to minimize self-supervised loss function (Equation 4) with new noisy images.

To resolve the noise misalignment issues, we shift the attention to the noise itself at the input level. In particular, we formulate a new unseen noise as deviation from the seen noise distribution:

$$\boldsymbol{e}^u = \boldsymbol{e}^s + \boldsymbol{\epsilon}^{s\to u}, \tag{5}$$

where $\boldsymbol{\epsilon}^{s\to u}$ represents how much $\boldsymbol{e}^u$ deviates from an arbitrary noise $\boldsymbol{e}^s$ sampled from $\mathcal{D}^s$. Thus, a new noisy image $\boldsymbol{y}^u$ can be seen as follows:

$$\begin{aligned} \boldsymbol{y}^u &= \boldsymbol{x}^u + \boldsymbol{e}^u & (6)\\ &= \boldsymbol{x}^u + \boldsymbol{e}^s + \boldsymbol{\epsilon}^{s\to u}. & (7) \end{aligned}$$

Given the formulation, we observe that we can mitigate the misalignment issues if we can adapt a given noisy image $\boldsymbol{y}^u$ to its translated counterpart noisy image $\boldsymbol{y}^{u\to s}$ with seen noise $\boldsymbol{e}^s \sim \mathcal{D}^s$, by removing the deviations $\boldsymbol{\epsilon}^{s\to u}$ as:

$$\begin{aligned} \boldsymbol{y}^{u\to s} &:= \boldsymbol{x}^u + \boldsymbol{e}^s & (8)\\ &= \boldsymbol{x}^u + \boldsymbol{e}^s + \boldsymbol{\epsilon}^{s\to u} - \boldsymbol{\epsilon}^{s\to u} & (9)\\ &= \boldsymbol{y}^u - \boldsymbol{\epsilon}^{s\to u}. & (10) \end{aligned}$$

Thus, our goal becomes to find a deviation offset $-\boldsymbol{\epsilon}^{s\to u}$ that we can add to a given noisy image $\boldsymbol{y}^u$ to adapt underlying noise towards noise a pretrained network is more familiar with. To this end, we add a learnable parameter $\boldsymbol{\phi}$ to a given noisy image $\boldsymbol{y}^u$ and then train $\boldsymbol{\phi}$ to approximate the deviation offset $-\boldsymbol{\epsilon}^{s\to u}$, as also illustrated in Figure 1:

$$\boldsymbol{y}^{u\to s} \approx \boldsymbol{y}^u + \boldsymbol{\phi} . \tag{11}$$

However, we do not have access to such deviation offset during the test phase, as new noise distribution is generally unknown and it would be also difficult to explicitly model seen noise distribution. To approximate an unknown $-\boldsymbol{\epsilon}^{s\to u}$ with $\boldsymbol{\phi}$, we train $\boldsymbol{\phi}$ to minimize a self-supervision loss function (Equation 4), under the assumption that loss function is minimized when an input noise becomes closer to seen noise distribution. The assumption is reasonable as a self-supervision loss function is a surrogate of Equation 2, which a pretrained network $f_{\boldsymbol{\theta}^*}$ is trained to minimize with seen noise distribution. Overall, our objective function to train $\boldsymbol{\phi}$ becomes:

$$\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\arg\min} \left\| f_{\boldsymbol{\theta}^*}(D_1(\boldsymbol{y}^u + \boldsymbol{\phi})) - D_2(\boldsymbol{y}^u + \boldsymbol{\phi}) \right\|_2^2 . \tag{12}$$

Note that we freeze the parameters of a pretrained denoising network $\boldsymbol{\theta}^*$ and only optimize $\boldsymbol{\phi}$. Then, finally, a clean image is estimated by

$$\hat{\boldsymbol{x}}^u = f_{\boldsymbol{\theta}^*}(\boldsymbol{y}^u + \boldsymbol{\phi}^*). \tag{13}$$

4. Experiments

We perform experiments on scenarios of test noisy images with different noise from the training set, demonstrating the effectiveness of our LAN framework. The implementation details and experimental settings are in Section 4.1. Then, we present the results in Section 4.2. The discussions on zero-shot denoising, computational efficiency, and the effects of noise adaptation are covered in Sections 4.3, 4.4, and 4.5, respectively.

4.1. Experimental setup

Dataset. For training denoising networks, we use one of widely used real-world noise dataset SIDD [1]. To simulate unseen new noise scenarios during the test phase, we utilize other real-world noise datasets with noise distribution different from SIDD: in this work, PolyU [48] and Nam [37]. We split the original PolyU images into 512 × 512 patches and crop them into 256 × 256. For Nam, we use patches splited into 256 × 256.

Models. We conduct experiments on prominent denoising networks, DnCNN [54], Restormer [53], and Uformer [46]. For Restormer and Uformer, we employ the parameters of SIDD-pretrained networks that are publicly available. On other hand, DnCNN does not have publicly available SIDD-pretrained network parameters. Thus, we train DnCNN from scratch using the SIDD dataset.

Evaluation. We measure PSNR and SSIM to evaluate the performance of SIDD-pretrained networks and our LAN framework on noisy images from PolyU and Nam datasets, against four alternative adaptation methods: adapting a whole network ('full-trainable'), adapting only the first layer ('first-layer'), adapting only the last layer ('last-layer'), and adapting a whole network via meta-learning using first-order MAML [14, 38] ('meta-learning'). For 'meta-learning', we initialize the network from SIDD pretrained weights. Then we train with 200 samples and validate with 100 samples from the SIDD training and validation set. Such test-time adaptation is conducted individually for each image, using only noisy images. The adaptation runs for up to 20 iterations and is optimized through Adam optimizer [21] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. At this time, the learning rate of LAN is 5e-4. For alternative adaptation schemes, we use 5e-6 for 'full-train', 5e-4 for 'first-layer' and 1e-4 for 'last-layer'. The 'meta-learning' is with 1e-5 learning rate on both adaptation and meta update. The learning rates are empirically found for stable convergence of models trained with self-supervision loss functions.

4.2. Experimental results

Quantitative results. We test the adaptation methods in two scenarios: SIDD → PolyU and SIDD → Nam. SIDD → PolyU means a network is pretrained on SIDD dataset and evaluated on PolyU dataset, while SIDD → Nam represents a scenario, where a network is pretrained on SIDD and evaluated on Nam dataset. To demonstrate the applicability of our framework, we also perform experiments with various network backbones: namely, DnCNN [54], Restormer [53], and Uformer [46] over different self-supervised losses such as ZS-N2N [35] and Nbr2Nbr [17]. The experimental results are displayed in Table 1.

The results demonstrate that LAN exhibits notable performance improvement even after performing adaptation for only 5 iterations. Furthermore, our framework LAN is shown to consistently outperform a 'full-trainable' method that adapts all parameters of a network to a given noisy image with the same self-supervision loss function (Equation 4) across problem settings and network backbones. For instance, under the setting of Restormer with Nbr2Nbr self-supervised loss on SIDD → NAM, our LAN framework improves the pretraining performance by 0.35dB, while a 'full-trainable' method improves by only 0.05dB. As a matter of fact, a 'full-trainable' method often results in performance degradation, let alone improve the performance of a pretrained network.

One may argue that fine-tuning all parameters of a network may lead to overfitting, hence the reason for the relatively low performance gain. It can also be argued that our method of directly adjusting the noise in the input image can prevent overfitting and have similar effects to finetuning only the first layer of a network. However, 'full-trainable' adaptation often leads to performance improvements. Furthermore, finetuning only a first layer (denoted as 'first-layer' in the table) results in inferior performance compared to 'full-trainable' and our method. The results suggest that our method does not have similar effects to finetuning only the first layer. In contrast to fine-tuning only the first layer, our method offers fine-grained noise adaptation on a per-pixel basis, rather than relying solely on convolution, which carries a significant inductive bias.

On the other hand, the 'full-trainable' adaptation can sometimes bring improvements even after 20 iterations, as it can be seen with Uformer network backbone adapted with ZS-N2N loss function on Nam dataset. Then, one may be curious as to whether 'full-trainable' adaptation can outperform LAN if adaptation is performed for longer iterations. As such, we plot the performance curve of 'full-trainable' and our LAN method as the number of iterations increases with Uformer network backbone adapted with ZS-N2N loss function on Nam dataset, as visualized in Figure 5. As shown in the figure, even after performing adaptation for longer iterations, 'full-trainable' method fails to achieve performance on par with our method and starts to worsen the performance after near 20 iterations.

Qualitative results. We display the qualitative results of a pretrained network, 'full-trainable' adaptation, and our LAN framework in Figure 3. The images are obtained with Uformer finetuned via ZS-N2N for 20 iterations on Nam dataset. We also visualize the adapted noisy image by our method. Interestingly, noise adaptation sometimes has better PSNR/SSIM than an original noisy image. Then, one may think that the performance of LAN may be because noise adaptation is introducing additional denoising process. However, the denoising performance improvement by noise adaptation is small, compared to pretrained network and our whole LAN framework. Furthermore, we observe that noise adaptation itself does not always give better PSNR than an original image, as noted in last two rows of the figure. Notable performance improvement brought by LAN just with noise adaptation at input image suggests that noise adaptation is not just additional denoising process.

4.3. Zero-shot denoising

One may argue that the performance degradation is expected with the full adaptation of a network when train noise distribution and new test noise distribution greatly differ. Another alternative to finetuning of an network would be to train a randomly initialized network on a new noisy image from scratch via self-supervision loss functions for blind denoising. In fact, ZS-N2N [35] is specifically designed for training a denoising network on a single noisy image with unknown noise. Thus, we demonstrate such zero-shot denoising performance with DnCNN, trained via ZS-N2N and Nbr2Nbr on each image in PolyU and Nam dataset, the results of which are displayed in Table 3. After training a network for more than 1K iterations as suggested in [35], the zero-shot training yields significantly poor performance, compared to not just LAN but also SIDD-pretrained model and other alternative adaptation methods. Despite the deviations and misalignment in noise, the results suggest the benefits of exploiting the knowledge of denoising tasks from a large training set. This is similar to domain adaptation, supporting our formulation.

Model	Method	Iter.	SIDD → PolyU		SIDD → Nam	
			PSNR$^{\uparrow}$ (dB) / SSIM$^{\uparrow}$		PSNR$^{\uparrow}$ (dB) / SSIM$^{\uparrow}$	
			ZS-N2N	Nbr2Nbr	ZS-N2N	Nbr2Nbr
DnCNN	pretrained	-	38.10 / 0.952		36.60 / 0.930	
	full-trainable	5	38.07 / 0.951	38.08 / 0.951	36.60 / 0.929	36.60 / 0.929
		10	38.04 / 0.950	38.06 / 0.951	36.59 / 0.928	36.60 / 0.928
		20	37.99 / 0.949	38.02 / 0.949	36.56 / 0.925	36.56 / 0.925
	first-layer	5	37.93 / 0.948	37.95 / 0.948	36.48 / 0.923	36.46 / 0.923
		10	37.76 / 0.943	37.83 / 0.945	36.29 / 0.915	36.28 / 0.915
		20	37.47 / 0.935	37.67 / 0.941	35.95 / 0.902	36.02 / 0.904
	last-layer	5	38.12 / 0.952	38.13 / 0.952	36.69 / 0.931	36.70 / 0.931
		10	38.13 / 0.952	38.14 / 0.952	36.75 / 0.931	36.77 / 0.931
		20	38.13 / 0.952	38.14 / 0.952	36.78 / 0.930	**36.81** / 0.930
	meta-learning	5	38.23 / **0.955**	38.23 / **0.956**	36.56 / 0.936	36.54 / **0.935**
		10	38.23 / **0.955**	38.25 / 0.955	36.66 / 0.934	36.65 / 0.934
		20	38.17 / 0.953	38.20 / 0.954	35.69 / 0.931	36.68 / 0.930
	LAN (Ours)	5	38.22 / 0.954	38.16 / 0.953	36.73 / 0.934	36.66 / 0.932
		10	**38.29 / 0.955**	38.22 / 0.954	**36.79** / 0.936	36.71 / 0.933
		20	**38.29 / 0.955**	**38.31 / 0.956**	36.78 / **0.938**	36.80 / **0.935**
Restormer	pretrained	-	39.03 / 0.966		38.03 / 0.951	
	full-trainable	5	39.09 / 0.966	39.04 / 0.965	38.14 / 0.952	38.07 / 0.951
		10	39.12 / 0.965	39.04 / 0.965	38.23 / 0.952	38.08 / 0.950
		20	39.14 / 0.965	38.98 / 0.964	38.35 / 0.953	38.05 / 0.948
	first-layer	5	39.04 / 0.965	39.00 / 0.965	38.12 / 0.951	38.07 / 0.950
		10	38.96 / 0.964	38.89 / 0.964	38.05 / 0.950	37.95 / 0.948
		20	38.74 / 0.961	38.66 / 0.961	37.65 / 0.943	37.52 / 0.941
	last-layer	5	39.07 / 0.965	39.08 / 0.965	38.09 / 0.951	38.10 / 0.951
		10	39.06 / 0.965	39.07 / 0.965	38.12 / 0.950	38.14 / 0.950
		20	39.02 / 0.964	39.03 / 0.964	38.12 / 0.948	38.14 / 0.948
	meta-learning	5	39.12 / 0.966	39.12 / 0.966	38.15 / 0.954	38.15 / 0.953
		10	39.18 / 0.966	39.13 / 0.966	38.34 / 0.954	38.21 / 0.952
		20	39.19 / 0.965	39.06 / 0.964	38.49 / 0.954	38.17 / 0.949
	LAN (Ours)	5	39.23 / 0.968	39.09 / 0.967	38.31 / 0.957	38.14 / 0.953
		10	**39.30 / 0.969**	39.14 / 0.967	38.58 / 0.961	38.25 / 0.955
		20	39.28 / **0.969**	**39.17 / 0.968**	**38.86 / 0.965**	**38.38 / 0.958**
Uformer	pretrained	-	38.93 / 0.965		37.55 / 0.950	
	full-trainable	5	39.01 / 0.964	38.96 / 0.964	37.80 / 0.950	37.72 / 0.948
		10	39.01 / 0.963	38.92 / 0.963	37.97 / 0.950	37.77 / 0.946
		20	38.91 / 0.961	38.77 / 0.961	38.07 / 0.948	37.67 / 0.942
	first-layer	5	38.89 / 0.965	38.85 / 0.965	37.75 / 0.952	37.69 / 0.950
		10	38.82 / 0.964	38.78 / 0.964	37.76 / 0.951	37.65 / 0.948
		20	38.71 / 0.962	38.69 / 0.963	37.71 / 0.946	37.54 / 0.943
	last-layer	5	38.98 / 0.965	38.99 / 0.965	37.68 / 0.949	37.69 / 0.950
		10	39.00 / 0.965	39.01 / 0.965	37.79 / 0.949	37.81 / 0.949
		20	39.01 / 0.965	39.01 / 0.965	37.90 / 0.949	37.92 / 0.949
	meta-learning	5	39.10 / 0.967	39.09 / **0.967**	37.77 / 0.957	37.87 / 0.955
		10	39.20 / 0.966	**39.11** / 0.966	38.26 / 0.957	**38.07** / 0.952
		20	39.11 / 0.964	38.97 / 0.963	**38.52** / 0.956	38.00 / 0.947
	LAN (Ours)	5	39.12 / 0.967	39.00 / 0.966	37.82 / 0.955	37.69 / 0.951
		10	**39.21 / 0.968**	39.05 / 0.966	38.09 / 0.960	37.83 / 0.953
		20	39.20 / **0.968**	39.10 / **0.967**	38.36 / **0.964**	38.02 / **0.956**

Table 1. **Quantitative comparison on denoising performance** for each combination of a denoising network backbone, an adaptation method, and a self-supervision loss on the real-world noise datasets (PolyU and Nam) after pretraining on another real-world noise dataset, SIDD.

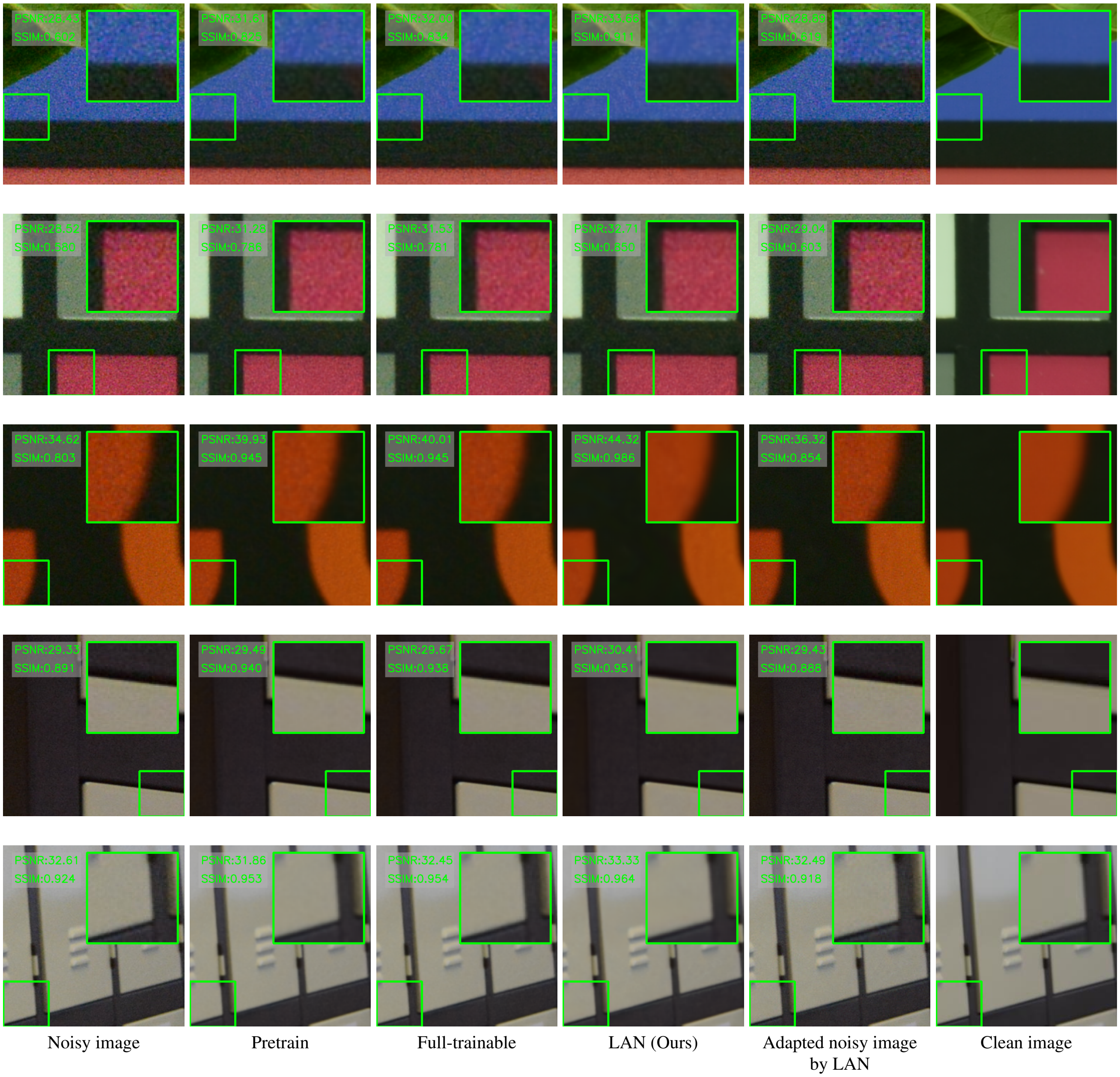


Figure 3. Qualitative comparisons among different adaptation methods. Images are obtained with SIDD-pretrained Uformer. Full-trainable and LAN (Ours) finetuned the pretrained network via ZS-N2N for 20 iterations on Nam (first three rows) and PolyU (last two rows).

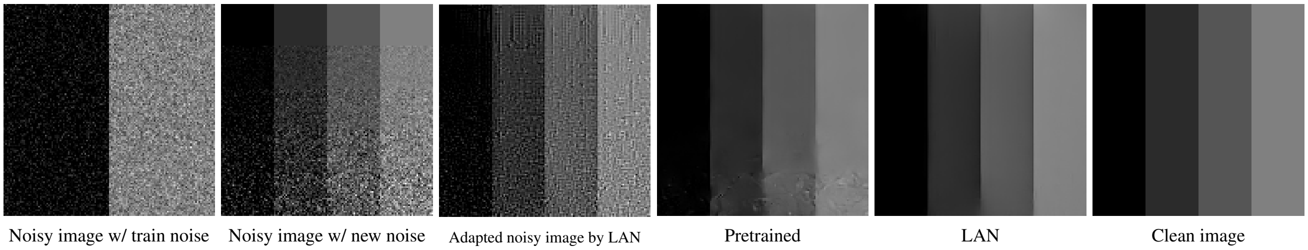


Figure 4. Visualization of synthetic noisy images. Noisy image with train noise is a noisy image that is used for pretraining a denoising network (DnCNN). Noisy image with new noise contain a new noise that is different from pretraining. Adapted noisy image by LAN is a result of noise adaptation of noisy image with new noise. We observe that noise in the adapted noisy image becomes more similar to noisy image with train noise. Particularly, we observe that noise has been added to the top of the image, where there was previously no noise. As a result, LAN helps achieve better denoising performance.

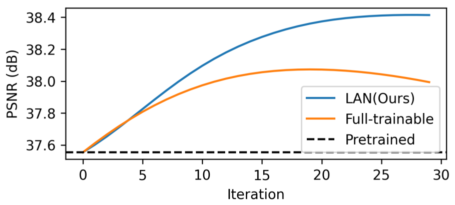


Figure 5. Plot of performance in PSNR over the number of adaptation iterations. Results are obtained with Uformer finetuned via ZS-N2N on Nam Dataset.

Model	Self-loss	LAN / Full-trainable	
		Time	Memory
Restormer	ZS-N2N	79.88 %	93.27%
	Nbr2Nbr	93.04 %	92.22%
Uformer	ZS-N2N	74.10 %	73.75%
	Nbr2Nbr	85.24%	74.21%

Table 2. The runtime and memory efficiency ratio of LAN (ours) to full-trainable based on an image size of 256 × 256.

4.4. Computational efficiency

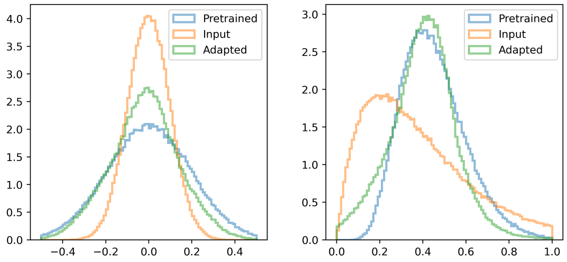
One may be concerned with the computational cost of our proposed LAN framework due to the introduction of a learnable parameter for each image pixel. However, our evaluations show that LAN is more efficient in memory and runtime including both adaptation and inference, especially for large networks like Restormer and Uformer, as displayed in Table 2. This is because fewer parameters need updating compared to a 'full-trainable' adaptation method. A limitation is that the number of trainable parameters depends on the input image size, which could reduce efficiency for large images. We plan to improve this in the future. Despite this, the notable performance improvement from our framework offers a promising research direction to explore for image denoising.

4.5. Effects of noise adaptation.

To better illustrate the effects of noise adaptation by LAN (Equation 11), we perform experiments with synthetic noises for clear visualization. Specifically, Gaussian $\sigma = 50$ or Gamma distribution $\theta = 0.5, k = 9$ noise is added to the single training data from BSDS500 [36] dataset to create synthetic-noisy-clean image pairs for pretraining the DnCNN. Figure 6 shows a histogram of noise from a synthetic training set (denoted as 'Pretrained'), a new noisy image (denoted as 'Input'), and an adapted noisy image (denoted as 'Adapted'). The figure demonstrates that our algorithm brings a new noise closer towards a noise expected by a denoising network. We also visualize such noisy images in Figure 4, where a plain image is used for better visualization. LAN is shown to try to adapt a new noisy image to contain similar noise used during pretraining. In particular, we observe newly added noise on top of adapted noisy image. The results illustrate that our noise adaptation is not just additional denoising.

Dataset	Self-loss	PSNR (dB)	SSIM
PolyU	ZS-N2N	31.47	0.875
	Nbr2Nbr	32.93	0.911
Nam	ZS-N2N	34.47	0.902
	Nbr2Nbr	35.39	0.923

Table 3. Zero-shot denoising performance with randomly initialized DnCNN, trained via ZS-N2N on each image in PolyU and Nam dataset.



(a) Adapted Gaussian noise

(b) Adapted gamma noise

Figure 6. Histogram of synthetic noise distributions. Adapted noise distribution (green) by LAN is shown to shift the new noise distribution (orange) to already seen noise distribution (blue).

5. Conclusion

In this work, we propose a new adaptation approach to handle unseen noise for image denoising. We focus on the fine-grained pixel-level misalignment issues between unseen noise in new noisy images and seen noise during the pretraining of a denoising network. In contrast to standard approaches of finetuning a model, we focus on adapting an input noisy image itself. To this end, we introduce a new denoising framework, named Learning-to-Adapt-Noise (LAN), that adds a new noisy image with a learnable offset that is trained to bring noise in a new noisy image closer to noise seen during the pretraining stage. The experimental results solidifies the motivation and effectiveness of noise adaptation by our proposed method. One limitation would be that the computation and resource complexities may grow with the size of input images, although LAN is more efficient in comparison to model adaptation for images of typical size 256×256. Nevertheless, we believe that our work brings interesting results and research discussions, and suggests a new research direction.

6. Acknowledgements

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022- 0-00156, Fundamental research on continual meta-learning for quality enhancement of casual videos and their 3D metaverse transformation), (No.2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)).

References

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

[2] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019. 2

[3] Abdelrahman Abdelhamed, Mahmoud Afifi, Radu Timofte, and Michael S Brown. Ntire 2020 challenge on real image denoising: Dataset, methods and results. In *CVPR*, 2020. 1

[4] Josue Anaya and Adrian Barbu. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. 1

[5] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, 2019. 1, 2

[6] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *ICML*, 2019. 2, 3

[7] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 2

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017. 3

[9] Ke-Chi Chang, Ren Wang, Hung-Jin Lin, Yu-Lun Liu, Chia-Ping Chen, Yu-Lin Chang, and Hwann-Tzong Chen. Learning camera-aware noise models. In *ECCV*, 2020. 2

[10] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021. 2

[11] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *CVPR*, 2021. 1

[12] 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. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigoldand Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 4

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[16] Agus Gunawan, Muhammad Adi Nugroho, and Se Jin Park. Test-time adaptation for real image denoising via meta-transfer learning. *arXiv preprint arXiv:2207.02066*, 2022. 2

[17] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *CVPR*, 2021. 2, 5

[18] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *CVPR*, 2021. 3, 4

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[20] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2n: Practical generative noise modeling for real-world denoising. In *ICCV*, 2021. 2

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[23] Shayan Kousha, Ali Maleky, Michael S Brown, and Marcus A Brubaker. Modeling srgb camera noise with normalizing flows. In *CVPR*, 2022. 2

[24] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *CVPR*, 2019. 2

[25] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4544–4553, 2020. 2

[26] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 2

[27] Seunghwan Lee and Tae Hyun Kim. Noisetransfer: Image noise generation with contrastive embeddings. In *ACCV*, 2022. 2

[28] Seunghwan Lee, Donghyeon Cho, Jiwon Kim, and Tae Hyun Kim. Self-supervised fast adaptation for denoising via meta-learning. *arXiv preprint arXiv:2001.02899*, 2020. 2

[29] 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 *CVPR*, 2022. 2

[30] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *ICML*, 2018. 2, 3

[31] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021. 2

[32] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. 2017. 2

[33] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. *arXiv preprint arXiv:1805.11145*, 2018. 2

[34] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009. 2

[35] Youssef Mansour and Reinhard Heckel. Zero-shot noise2noise: Efficient image denoising without any data. In *CVPR*, 2023. 2, 3, 4, 5

[36] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 416–423. IEEE, 2001. 8

[37] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016. 1, 4

[38] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 4

[39] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 1

[40] 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. 2

[41] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 2

[42] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 3

[43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 2

[44] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 2

[45] Jiachuan Wang, Shimin Di, Lei Chen, and Charles Wang Wai Ng. Noise2info: Noisy image to information of noise for self-supervised image denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16034–16043, 2023. 2

[46] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022. 2, 4, 5

[47] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *NeurIPS*, 2012. 2

[48] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 1, 4

[49] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8978–8987, 2021. 2

[50] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020. 2

[51] 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 *CVPR*, 2020. 2

[52] 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 *ECCV*, 2020. 1

[53] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 2, 4, 5

[54] 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, 4, 5

[55] 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. 2

[56] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 2