Zero-Shot Noise2Noise: Efficient Image Denoising without any Data

Youssef Mansour and Reinhard Heckel

Technical University of Munich and Munich Center for Machine Learning

Munich, Germany

y.mansour@tum.de, reinhard.heckel@tum.de

Abstract

Recently, self-supervised neural networks have shown excellent image denoising performance. However, current dataset free methods are either computationally expensive, require a noise model, or have inadequate image quality. In this work we show that a simple 2-layer network, without any training data or knowledge of the noise distribution, can enable high-quality image denoising at low computational cost. Our approach is motivated by Noise2Noise and Neighbor2Neighbor and works well for denoising pixel-wise independent noise. Our experiments on artificial, real-world camera, and microscope noise show that our method termed ZS-N2N (Zero Shot Noise2Noise) often outperforms existing dataset-free methods at a reduced cost, making it suitable for use cases with scarce data availability and limited compute.

1. Introduction

Image denoising is the process of removing distortions from images, to enhance them visually and to reconstruct fine details. The latter is especially important for medical images, where fine details are necessary for an accurate diagnosis.

Current state-of-the-art image denoising techniques rely on large data sets of clean-noisy image pairs and often consist of a neural network trained to map the noisy to the clean image. The drawbacks of dataset based methods are that data collection, even without ground truths, is expensive and time-consuming, and second, a network trained on dataset suffers from a performance drop if the test images come from a different distribution of images. These drawbacks motivate research in dataset-free methods.

All current zero-shot models are either suitable only for specific noise distributions and need previous knowledge of the noise level [7, 20], require a lot of compute (time, memory, GPU) to denoise an image [24], have

poor denoising quality [28], or do not generalise to different noise distributions or levels [15, 24]. We propose a method that builds on the recent Noise2Noise [17] and Neighbour2Neighbour [12] papers and aims to circumvent these issues to reach a good trade-off between denoising quality and computational resources. We make only minimal assumptions on the noise statistics (pixel-wise independence), and do not require training data. Our method does not require an explicit noise model, and is therefore suitable for various noise types and can be employed when the noise distribution or level are unknown. The only assumption we make about the noise is that it is unstructured and has zero mean.

In a nutshell, we convolve the noisy test image with two fixed filters, which yields two downsampled images. We next train a lightweight network with regularization to map one downsampled image to the other. Our strategy builds on the recent Noise2Noise [17] and Neighbour2Neighbour [12] papers, however we take those methods one step further by enabling denoising without any training data. Even with an extremely small network and without any training data, our method achieves good denoising quality and often even outperforms large networks trained on datasets.

The key attributes of our work are as follows:

- **Compute.** Dataset free neural network based algorithms [24, 28] require solving an optimization problem involving millions of parameters to denoise an image. The huge parameter count requires large memory storage, advanced GPUs, and long denoising times. In this work we show that our method, that utilizes a simple 2 layer network, with only 20k parameters, can often outperform networks with millions of parameters while reducing the computational cost significantly and being easily executable on a CPU.

- **Generalisation.** Existing zero-shot methods often to do not generalise well. For example, BM3D [7], a classical denoising algorithm does not generalize well to non-Gaussian noise, and blind spot net-
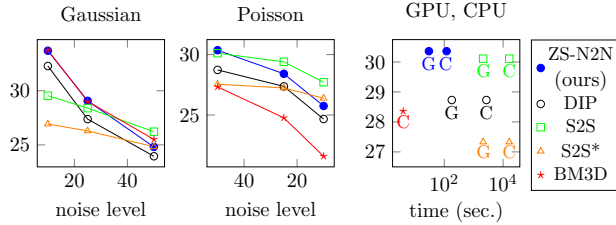
Figure 1. Left and middle plots: PSNR scores for Gaussian and Poission denoising for different noise levels. Note BM3D's poor performance on Poisson compared to Gaussian noise. Right plot: Time required in seconds to denoise one 256×256 colour image on CPU and GPU, tested on Poisson noise with $\lambda = 50$. Except for BM3D, all methods have shorter times on GPU. Only S2S in some cases outperforms our method, however it is about 100 times slower. S2S* denotes the ensemble free version of S2S.

works [15] [24] (discussed later in detail) fail to denoise well in the regime of low noise level. Extensive experiments on different noise distributions and noise levels show that our proposed approach can generalise better to different conditions better than existing methods.

In summary, our proposed method is dataset and noise model-free, and achieves a better trade-off between generalization, denoising quality, and computational resources compared to existing zero-shot methods, as displayed in Figure 1. We compare to the standard zero shot baselines, including BM3D, and the recent neural network-based algorithms DIP [28] and S2S [24]. Only BM3D is faster than our method but achieves poor results on non-Gaussian noise. Only S2S sometimes outperforms our method, but is orders of magnitude slower, often fails on low noise levels [14], and requires ensembling to achieve acceptable performance.

2. Related Work

Supervised methods achieve state-of-the-art performance by training a network end-to-end to map a noisy image to a clean one. Networks that work well are CNNs [3, 32], vision transformers [19], or MLP based architectures [21, 27].

Noise2Noise [17] yields excellent performance from training on two noisy images of the same static scene, without any ground truth images. Given that the noise has zero mean, training a network to map one noisy image to another noisy image of the same scene performs as well as mapping to the ground truth. While having access to a pair of noisy images of the same scene is in practice hard to achieve, the Noise2Noise method

has motivated further research in self-supervised methods [12] that require only single noisy images.

Self-supervised methods are trained on datasets consisting of only noisy images. Noise2Void [15] and Noise2Self [2] are two blind spot prediction approaches for image denoising. Given a set of noisy images $\{\mathbf{y}^i\}_1^n$, The idea is to minimize the loss $\frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_\theta(M^i(\mathbf{y}^i)), \mathbf{y}^i)$, where $\mathcal{L}$ is a loss function, f_θ is a network, and M^i is an operator that masks some pixels, hence the name blind spot. Assuming that the neighbouring pixels of a clean image are highly correlated, and that the noise pixels are independent, a network trained to reconstruct a masked pixel, can only predict the signal value from the neighbouring visible pixels, but not the noise.

Recently, several works [4, 26, 34] attempted to use Stein's unbiased risk estimator for Gaussian denoising. Such methods work well only for Gaussian noise and require the noise level to be known in advance. A more general framework is Noisier2Noise [23] which works for any noise distribution, but the distribution must be known in advance.

The newly proposed Neighbour2Neighbour [12] builds on the Noise2Noise [17] method, where the assumptions are that the noise has zero mean and is pixel-wise independent. Neighbour2Neighbour extends Noise2Noise by enabling training without noisy image pairs. It does so by sub-sampling single noisy images to create pairs of noisy images, where Noise2Noise can be applied.

Zero-Shot/ Dataset free Methods. Most similar to our work is Noise2Fast [18], which also builds on Noise2Noise and Neighbour2Neighbour to achieve dataset-free denoising. However, the method is only evaluated on grayscale images, uses a relatively large network, and requires an early stopping criterion. Our work improves on Noise2Fast by easily denoising grayscale or RGB images, and designing a consistency loss that alleviates the need to early stop. Moreover, we use a much smaller network which saves compute. Specifically, our network is twelve times smaller and a forward pass through it is seven times faster. To the best of our knowledge, our work is the first to utilize a small 2-layer network and achieve competitive quality for image restoration. We show that on grayscale images, our method despite achieving similar scores to Noise2Fast [18], produces better quality images. This is likely due to Noise2Fast dropping pixel values when downsampling, whereas our method always keeps all information retained.

Besides this work, classical non-learning-based methods, such as BM3D [7] and Anscombe [20], work

well for Gaussian and Poisson noise, respectively, and require the noise level as an input.

DIP (Deep Image Prior) [28] and its variants such as the Deep Decoder [11] build on the fact that CNNs have an inductive bias towards natural images, in that they can fit natural images much faster than noise. Therefore, a network trained, with early stopping, to map a random input to the noisy image will denoise the image. The denoising performance of DIP is often poor, and is dependent on the number of training epochs, which is hard to determine in advance.

Self2Self [24] utilizes the idea of the blind spot networks (reconstructing masked pixels) on a single image, but with dropout ensembling. However, this method is not computationally efficient, in that it requires long durations to denoise an image. According to the authors, it takes 1.2 hours to denoise one 256×256 image on a GPU. Compared to other blind spot networks, Self2Self achieves significantly better denoising scores, since it relies on ensembling, i.e., averaging the output of several networks. However, ensemble learning over smoothens the image, causing a loss of some details, despite the improvement in PSNR scores [8].

Similar to almost all supervised and self-supervised methods, both Self2Self and DIP use a UNet [25] or a variant of it as the backbone network in their architectures. A UNet typically has millions of parameters, making it unsuitable for compute limited applications. Our work departs from this scheme, by designing a shallow and simple network with few parameters.

3. Method

Our method builds on the Noise2Noise [17], for training a network on pairs of noisy images, and the Neighbour2Neighbour (NB2NB) [12], which generates such pairs from a single noisy image. Our main idea is to generate a pair of noisy images from a single noisy image and train a small network only on this pair. We start with a brief summary of Noise2Noise and then introduce our method.

3.1. Background: Noise2Noise and Neighbour2Neighbour

Supervised denoising methods are typically neural networks $f_{\boldsymbol{\theta}}$ that map a noisy image $\mathbf{y}$ to an estimate $f_{\boldsymbol{\theta}}(\mathbf{y})$ of the clean image $\mathbf{x}$. Supervised denoising methods are typically trained on pairs of clean images $\mathbf{x}$ and noisy measurements $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{e}$ is noise. We refer to supervised denoising as Noise2Clean (N2C).

Neural networks can also be trained on different noisy observations of the same clean image. Noise2Noise (N2N) [17] assumes access to a set of pairs of noisy images $\mathbf{y}_1 = \mathbf{x} + \mathbf{e}_1, \mathbf{y}_2 = \mathbf{x} + \mathbf{e}_2,$

where $\mathbf{e}_1, \mathbf{e}_2$ are independent noise vectors. A network $f_{\boldsymbol{\theta}}$ is then trained to minimize the empirical risk $\frac{1}{n} \sum_{i=1}^{n} \left\| f_{\boldsymbol{\theta}}(\mathbf{y}_1^i) - \mathbf{y}_2^i \right\|_2^2$. This makes sense, since in expectation over such noisy instances, and assuming zero mean noise, training a network in a supervised manner to map a noisy image to another noisy image is equivalent to mapping it to a clean image i.e.,

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}\left[\|f_{\boldsymbol{\theta}}(\mathbf{y}_1) - \mathbf{x}\|_2^2\right] = \arg\min_{\boldsymbol{\theta}} \mathbb{E}\left[\|f_{\boldsymbol{\theta}}(\mathbf{y}_1) - \mathbf{y}_2\|_2^2\right].$$
(1)

The proof is given in the supplementary material.

In theory N2N training reaches the same performance as N2C training if the dataset is infinitely large. In practice, since the training set is limited in size, N2N falls slightly short of N2C. For example, N2N training with a UNet on 50k images gives a performance drop of only about 0.02 dB compared to N2C with a UNet.

Despite the great performance of N2N, its usability is often limited, since it is difficult to obtain a pair of noisy images of the same static scene. For instance, the object being captured might be non-static, or the lighting conditions change rapidly.

Neighbour2Neighbour (NB2NB) [12] extends N2N and allows training only on a set of single noisy images, by sub-sampling a noisy image to create a pair of noisy images. Similar to N2N, NB2NB exhibits strong denoising performance when trained on many images.

3.2. Zero-Shot Noise2Noise

Our work extends Noise2Noise [17] and Neighbour2Neighbour [12] by enabling training on only one single noisy image. To avoid overfitting to the single image, we use a very shallow network and an explicit regularization term.

Almost all self- or un-supervised denoising methods, including ours, rely on the premise that a clean natural image has different attributes than random noise. As shown in [12], a noisy image can be decomposed into a pair of downsampled images. Based on the premise that nearby pixels of a clean image are highly correlated and often have similar values, while the noise pixels are unstructured and independent, the downsampled pair of noisy images has similar signal but independent noise. This pair can therefore serve as an approximation of two noisy observations of the same scene, where one observation is used as the input, and the other as the target, as in N2N.

Our approach is to first decompose the image into a pair of downsampled images and second train a lightweight network with regularization to map one downsampled image to the other. Applying the so-trained network to a noisy image yields the denoised image. We first explain how we generate the downsampled images, and then how we fit the network.

Image Pair Downsampler The pair downsampler takes as input an image $\mathbf{y}$ of size $H \times W \times C$ and generates two images $D_1(\mathbf{y})$ and $D_2(\mathbf{y})$, each of size $H/2 \times W/2 \times C$. The downsampler generates those images by dividing the image into non-overlapping patches of size 2×2, taking an average of the diagonal pixels of each patch and assigning it to the first low-resolution image, then the average of the anti-diagonal pixels and assigning it to the second low-resolution image. See Figure 2 for an illustration of the pair downsampler.

The downsampler is implemented with convolutions as follows. The first low-resolution image is obtained by applying a 2D convolution with stride two and fixed kernel $\mathbf{k_1} = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}$ to the original image as $D_1(\mathbf{y}) = \mathbf{y} \circledast \mathbf{k_1}$, and the second image is obtained by applying a 2D convolution with stride two and fixed kernel $\mathbf{k_2} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$ to the original image as $D_2(\mathbf{y}) = \mathbf{y} \circledast \mathbf{k_2}$. The convolutions are implemented channel-wise and therefore the downsampling scheme is applicable to any arbitrary number of input channels.
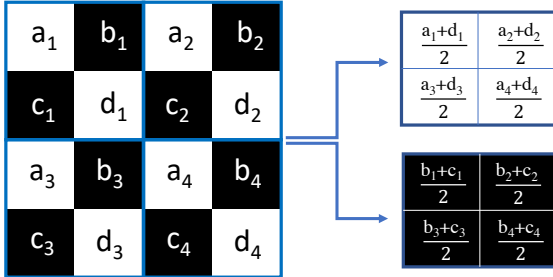

Figure 2. The Image Pair Downsampler decomposes an image into two images of half the spatial resolution by averaging diagonal pixels of 2×2 non-overlapping patches. In the above example the input is a 4×4 image, and the output is two 2×2 images.

Zero-shot-image denoising method. Given a test image $\mathbf{y}$ to denoise, our method is conceptually similar to first fitting a small image-to-image neural network $f_{\boldsymbol{\theta}}$ to map the first downsampled image $D_1(\mathbf{y})$ to the second one, $D_2(\mathbf{y})$ by minimizing the loss

$$\mathcal{L}(\boldsymbol{\theta}) = \|f_{\boldsymbol{\theta}}(D_1(\mathbf{y})) - D_2(\mathbf{y})\|_2^2. \quad (2)$$

Once we fitted the network, we can apply it to the original noisy observation to estimate the denoised image as $\hat{\mathbf{x}} = f_{\hat{\boldsymbol{\theta}}}(\mathbf{y})$.

However, our experiments showed that residual learning, a symmetric loss, and an additional consistency-enforcing term are critical for good performance. We next explain the elements of our loss function. In residual learning, the network is optimized to fit

the noise instead of the image. The loss then becomes

$$\mathcal{L}(\boldsymbol{\theta}) = \|D_1(\mathbf{y}) - f_{\boldsymbol{\theta}}(D_1(\mathbf{y})) - D_2(\mathbf{y})\|_2^2. \quad (3)$$

Following [6], where a symmetric loss was used in the context of self-supervised pretraining of a siamese network, we additionally adopt a symmetric loss, which yields the residual loss:

$$\mathcal{L}_{\text{res.}}(\boldsymbol{\theta}) = \frac{1}{2}\Big(\|D_1(\mathbf{y}) - f_{\boldsymbol{\theta}}(D_1(\mathbf{y})) - D_2(\mathbf{y})\|_2^2 + \|D_2(\mathbf{y}) - f_{\boldsymbol{\theta}}(D_2(\mathbf{y})) - D_1(\mathbf{y})\|_2^2 \Big). \quad (4)$$

In addition, we enforce consistency by ensuring that first denoising the image $\mathbf{y}$ and then downsampling it, is similar to what we get when first downsampling $\mathbf{y}$ and then denoising it, i.e., we consider a loss of the form:

$$\mathcal{L}(\boldsymbol{\theta}) = \|D(\mathbf{y}) - f_{\boldsymbol{\theta}}(D(\mathbf{y})) - D(\mathbf{y} - f_{\boldsymbol{\theta}}(\mathbf{y}))\|_2^2. \quad (5)$$

Again adopting a symmetric loss, the consistency loss becomes:

$$\mathcal{L}_{\text{cons.}}(\boldsymbol{\theta}) = \frac{1}{2}\Big(\|D_1(\mathbf{y}) - f_{\boldsymbol{\theta}}(D_1(\mathbf{y})) - D_1(\mathbf{y} - f_{\boldsymbol{\theta}}(\mathbf{y}))\|_2^2 + \|D_2(\mathbf{y}) - f_{\boldsymbol{\theta}}(D_2(\mathbf{y})) - D_2(\mathbf{y} - f_{\boldsymbol{\theta}}(\mathbf{y}))\|_2^2 \Big). \quad (6)$$

Note that for the residual loss, the network only has the downsampled images as input. Only in the consistency loss, the network gets to see the image in full spatial resolution. Including the consistency loss enables better denoising performance and helps to avoid overfitting. It can therefore be seen as a regularizing term.

In summary, we minimize the loss $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{res.}}(\boldsymbol{\theta}) + \mathcal{L}_{\text{cons.}}(\boldsymbol{\theta})$ using gradient descent, which yields the network parameters $\hat{\boldsymbol{\theta}}$. With those, we estimate the denoised image as $\hat{\mathbf{x}} = \mathbf{y} - f_{\hat{\boldsymbol{\theta}}}(\mathbf{y})$. Note that only the network parameters $\boldsymbol{\theta}$ are optimized during the gradient descent updates, since the downsampling operations D_1 and D_2 are fixed. Convergence typically requires 1k to 2k iterations, which thanks to using a lightweight network takes less than half a minute on a GPU and around one minute on a CPU.

Network Many supervised and self-supervised methods use a relatively large network, often a UNet [25]. Instead, we use a very simple two-layer image-to-image network. It consists of only two convolutional operators with kernel size 3×3 followed by one operator of 1×1 convolutions. This network has about 20k parameters, which is small compared to typical denoising networks. An exact comparison of the network sizes can be found in section 4.4. There are no normalization or pooling layers. The low parameter count and simple structure enables fast denoising even when deployed on a CPU.

In the ablation studies we show that using a UNet instead of a lightweight network leads to overfitting and much worse denoising performance.

4. Experiments

We compare our denoising algorithm (ZS-N2N) to several baselines. The baselines include dataset based methods, as well as other zero-shot methods. For the dataset based methods, we include both supervised (with clean images) and self-supervised (only noisy images) methods. We test all methods on artificial and real-world noise. We provide ablation studies in the supplementary material.

The results highlight the dependency of dataset based methods on the dataset they are trained on and suggest that given a small training set, they are outperformed by dataset free ones. Furthermore, the experiments show that methods based on noise models achieve good performance for the specific noise model, but do not generalise to other distributions.

Concerning the dataset and noise model free methods, our proposed method is either on par or better than other baselines on Gaussian, Poisson, and real world camera and microscope noise. Our method only falls short of Self2Self [24] on high noise levels, however, it requires only $\frac{1}{200}$ of the denoising time of Self2Self and 2% of it's memory. Moreover, Self2self's performance on low noise levels is insufficient. Therefore, considering denoising quality, generalistion, and computational resources, our method achieves a better trade-off compared to existing methods as shown in Figure 1.

4.1. Baselines

We compare to Noise2Clean (N2C) with a UNet, which is the current state-of-the-art denoising algorithm. There exits several other networks that perform on par with the UNet, such as DnCNN [32] and RED30 [22], but the UNet is orders of magnitude faster, since it is not very deep, and has a multi-resolution structure. The UNet is therefore the standard choice in all recent denoising papers [12, 15, 17, 23].

For the self-supervised methods, we compare to Neighbour2Neighbour (NB2NB) [12] and Noise2Void (N2V) [15]. We exclude the methods that require an explicit noise model, such as [4, 16, 23, 34], since these methods work well on synthetic denoising tasks for the given noise distribution, but fail to generalize to unknown noise distributions or real-world noise [12, 30]. This is due to the fact that the synthetic noise is insufficient for simulating real camera noise, which is signal-dependent and substantially altered by the camera's imaging system.

Regarding the zero-shot methods, which are most similar to ours, we compare to the deep learning based algorithms: DIP [28] and Self2Self (S2S) [24], and also to the classical algorithm: BM3D [7]. Note that apart of our method (and BM3D), all baselines use a U-Net or a variation of it as a denoising backbone.

The performance of DIP is very sensitive to the number of gradient descent steps. We used the ground truth images to determine the best early stopping iteration. The DIP results can therefore be seen as an over optimistic performance of the method. For a fair comparison, we report the results of the best performing model for the other baselines. A comparison of the sensitivity of the methods to the number of optimization steps can be found in the supplementary material.

The original implementation of S2S uses an ensemble of multiple networks, i.e, averaging the outputs of several networks. All other baselines do not utilize ensembling or averaging. For a fair comparison, we additionally report the results of S2S without any ensembling, which we denote by S2S*. S2S denotes the original implementation with an ensemble of 50 networks.

4.2. Synthetic Noise

The dataset based methods (N2C, NB2NB, N2V) are trained on 500 colour images from ImageNet [10]. All methods are tested on the Kodak24 [1] and McMaster18 [13] datasets. All training and test images are center-cropped to patches of size 256×256.

We examine Gaussian and Poisson noise with noise levels σ and λ respectively. We consider the fixed noise levels σ, λ= 10, 25, 50. The σ values for Gaussian noise correspond to pixel values in the interval [0,255], while the λ values for Poisson noise correspond to values in the interval [0,1].

For the dataset based methods, we also consider blind denoising during training with the range of noise levels $\sigma, \lambda \in [10, 50]$. During training, a σ, λ value is sampled uniformly from the given range for each image in each training epoch, unlike the fixed noise levels, where all training images are contaminated with the same noise level. Blind denoising is what is used in practice, since an exact noise level is typically not given, but rather a range of noise levels.

In table 1, we present the denoising performance of the different methods. For the dataset based methods, σ, λ is known, denotes that the network trained on that exact noise level is used for testing, while unknown denotes the blind denoising, where the network trained on the range of noise levels [10,50] is used for testing. BM3D requires as input the value of the noise level. For Gaussian denoising the known σ value was used, while for Possion denoising the noise level was estimated us-

---

ing the method in [5]. Note that ZS-N2N, DIP, and S2S do not utilize any prior information on the noise distribution or level.

As seen from the results, the dataset based methods often fall slightly short of the dataset free methods. This is due to the fact that they were only trained on 500 images, whereas they reach good performance when trained on larger datasets. In the supplementary material, we show that when N2C is trained on 4000 images, it outperforms all other baselines and its performance can keep improving with more training data. Another drawback of dataset based methods is that they are sensitive to the data they are trained on. They experience a performance drop when trained on a range of noise levels as opposed to a specific noise level as the test set.

Regarding the zero-shot methods, DIP exhibited worse scores in all simulations. BM3D is tailored to work well for Gaussian denoising, where the exact noise variance is known and required as input. However, its performance dropped for Poisson noise, where the noise level was estimated.

ZS-N2N and S2S do not rely on a specific noise model and therefore work consistently well for both Gaussian and Poisson noise. However, S2S suffers from at least two drawbacks. The first is it heavily relies on ensembling to achieve good scores as seen by comparing the results of S2S with S2S*. Despite improving the scores, ensembling oversmoothens the image causing a loss in some visual features [8]. Note that all other baselines are ensemble free. The second drawback is that it performs worse than all other baselines on low noise levels, as seen in the Gaussian noise with $\sigma = 10$.

Considering that DIP performs poorly, that BM3D only works well for Gaussian noise, and that S2S's performance without ensembling and on low noise levels is unsatisfactory, our method, ZS-N2N is the only dataset free denoising algorithm that performs well on different noise distributions and levels.

4.3. Real-World Noise

Camera noise: Following [24], we evaluate on the PolyU dataset [29] which consists of high-resolution images from various scenes captured by 5 cameras from the 3 leading brands of cameras: Canon, Nikon, and Sony. We also consider the SIDD [1], which consists of images captured by several smartphone cameras under different lighting conditions and noise patterns.

Since the computational cost for running S2S is high, we randomly choose 20 images from both datasets to test on. The SIDD validation set has images of size 256×256. For consistency, we center-crop the PolyU images to patches of size 256×256. The results are shown in table 2. All methods perform similarly except for BM3D and the ensemble free version of S2S, which exhibit a notable performance drop.

Dataset	ZS-N2N	DIP	S2S	S2S*	BM3D
PolyU	36.92	**37.07**	<u>37.01</u>	33.12	36.11
SIDD	<u>34.07</u>	**34.31**	33.98	30.77	28.19

Table 2. Denoising PSNR in dB on real world camera noise.

Microscope noise: We additionally evaluate on the Fluorescence Microscopy dataset [33], which contains real grayscale fluorescence images obtained with commercial confocal, two-photon, and wide-field microscopes and representative biological samples such as cells, zebrafish, and mouse brain tissues. We pick random images from the test set to test on. We also compare to Noise2Fast (N2F) [18], for which code for denoising grayscale is available. The results are depicted in table 3.

Image	Photon BPAE	Photon Mice	Confocal BPAE	Average
ZS-N2N	30.73	<u>31.42</u>	<u>35.85</u>	**32.67**
DIP	29.22	30.01	35.51	<u>31.58</u>
S2S	<u>30.90</u>	**31.51**	31.01	31.14
S2S*	29.49	29.99	29.54	29.67
BM3D	27.19	29.48	33.23	29.97
N2F	**30.93**	31.07	**36.01**	**32.67**

Table 3. PSNR in dB on real world microscope noise.

Our method and Noise2Fast achieve similar scores and slightly outperform the other baselines. Despite the similarity in scores, when inspecting the denoised images visually, we see differences: Our method produces visually sharper images and preserves slightly more details, while the Noise2Fast images are relatively smooth. This is most noticeable on images with fine details, such as MRI images, see Figure 3 for a knee image from the fastMRI dataset [31]. The blurriness in the Noise2Fast images is likely due to the downsampling scheme used, which drops some pixel values, and the ensembling performed to obtain the final image estimate, which oversmoothens the image [8]. Our method, on the other hand, preserves all pixel values during downsampling, and is ensemble free.

4.4. Computational Efficiency

In this section we focus on the computational efficiency. We consider the denoising time and the memory requirements represented by the number of network parameters. Since in some applications a GPU is not available [9], we additionally consider the denoising time on a CPU. The GPU tested is Quadro RTX 6000 and the CPU is Intel Core i9-9940X 3.30GHz.

Table 1 header and data:

Noise		Method	σ known?	Kodak24			McMaster18		
				$\sigma=10$	$\sigma=25$	$\sigma=50$	$\sigma=10$	$\sigma=25$	$\sigma=50$
Gaussian	dataset-based	*N2C*	yes	33.45	28.27	25.47	33.03	28.46	**25.86**
			no	32.16	28.18	24.45	31.97	28.26	24.78
		NB2NB	yes	33.01	27.90	25.02	32.63	28.01	<u>25.25</u>
			no	31.79	27.80	24.15	31.19	27.85	23.95
		N2V	yes	30.19	26.21	24.07	30.95	26.50	23.94
			no	28.95	26.03	23.19	29.64	26.31	22.67
	dataset-free	ZS-N2N (ours)	-	<u>33.69</u>	**29.07**	24.81	<u>34.21</u>	<u>28.80</u>	24.02
		DIP	-	32.28	27.38	23.95	33.07	27.61	23.03
		S2S	-	29.54	28.39	**26.22**	30.78	28.71	25.03
		S2S*	-	26.93	26.29	24.83	27.64	26.48	23.79
		BM3D	yes	**33.74**	<u>29.02</u>	<u>25.51</u>	**34.51**	**29.21**	24.51

Noise		Method	λ known?	$\lambda=50$	$\lambda=25$	$\lambda=10$	$\lambda=50$	$\lambda=25$	$\lambda=10$
Poisson	dataset-based	*N2C*	yes	<u>29.42</u>	27.49	<u>26.25</u>	<u>29.89</u>	28.20	26.42
			no	28.92	27.14	23.13	28.62	27.51	24.32
		NB2NB	yes	29.19	27.01	25.71	29.41	27.79	25.95
			no	28.53	26.88	23.60	28.03	27.66	24.58
		N2V	yes	27.73	25.55	23.77	27.86	25.65	23.47
			no	27.04	25.28	21.93	26.34	25.52	22.07
	dataset-free	ZS-N2N (ours)	-	**29.45**	<u>27.52</u>	24.92	**30.36**	<u>28.41</u>	25.75
		DIP	-	27.51	25.84	23.81	28.73	27.37	24.67
		S2S	-	28.89	**28.31**	**27.29**	<u>30.11</u>	**29.40**	**27.71**
		S2S*	-	26.75	26.40	25.63	27.55	27.24	26.39
		BM3D	no	28.36	26.58	24.20	27.33	24.77	21.59

Table 1. PSNR scores in dB for Gaussian and Poisson denoising. Best result is in **bold**, second best result is <u>underlined</u>. The dataset based methods are *italicized*. Note DIP's mediocre scores and BM3D's performance drop between Gaussian and Poission noise. S2S has significantly lower scores in low noise as seen with $\sigma = 10$ and its ensemble free version S2S* has inadequate performance. Denoised samples can be found in the supplementary material.

In table 4 we display the time required to denoise one colour image of size 256×256 at inference, as well as the total number of trainable parameters of a model. The dataset based methods are trained for long durations, but after training, the network parameters are fixed, and inference is almost instantaneous, since it is just a forward pass through the model. The time taken for denoising is therefore negligible compared to the zero-shot methods, whose parameters are optimized for each test image separately.

In the original implementation of S2S, the authors report a denoising time of 1.2 hours for a 256×256 colour image on GPU. However, we noticed that only half of the gradient update iterations are needed for convergence. We therefore report only half of their GPU time.

Concerning the denoising time, dataset based methods are the fastest, since a forward pass through a fixed network requires only milli seconds. Regarding the deep learning based zero-shot methods, ZS-N2N is significantly more computationally efficient. Specifically, on CPU it is 200 times and 35 times faster than S2S and

DIP respectively and has only 2% and 1% of their memory requirements. Only the classical BM3D is computationally more efficient than ZS-N2N.

4.5. Discussion

Dataset based methods typically achieve state-of-the-art results but our experiments manifested two of their shortcomings: They don't perform well when trained on small datasets, and the performance drops when the test data differs from the training data, as seen by varying the noise levels. This highlights the importance of dataset free denoising algorithms.

Methods that rely on an explicit model of the noise distribution such as Noisier2Noise [23] and Anscombe [20] or those tailored to work well for specific distributions such as BM3D, do not generalize well to other distributions. Their performance therefore degrades when the noise distribution is unknown, or the noise level must be estimated. This has been manifested by BM3D's competitive performance on Gaussian noise, but its failure to keep up with the other baselines on Poission and real world noise. These findings highlight the advantage
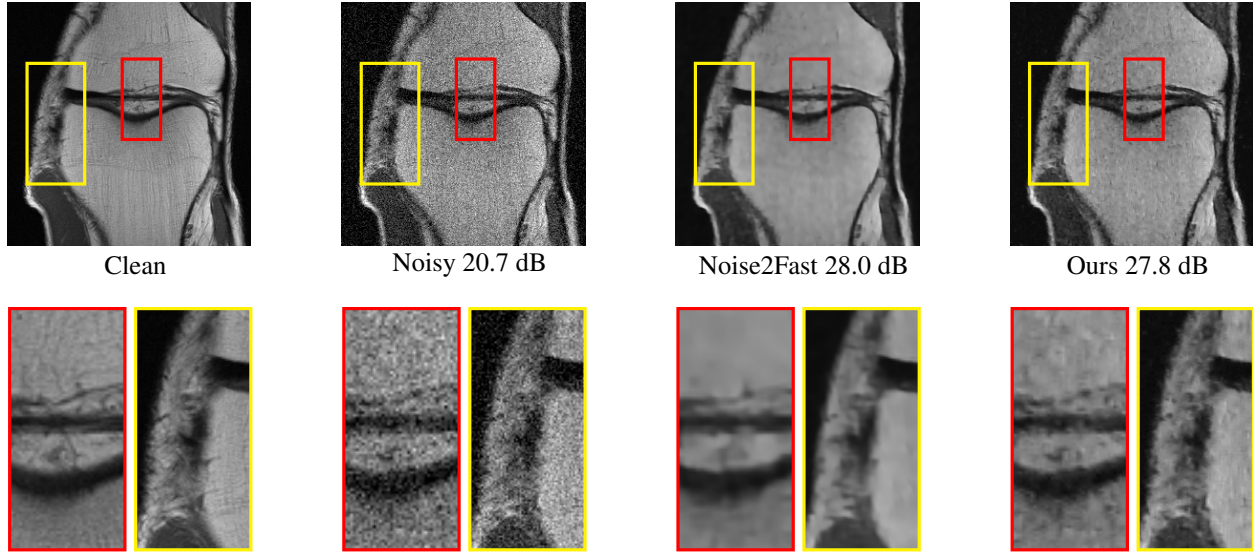
Figure 3. Visual comparison between our method and Noise2Fast for denoising Gaussian noise on a knee MRI. Both methods achieve similar PSNR, but notice how the center and left edge are blurry and oversmooth in Noise2Fast. Our method produces a sharper image with less loss of details.

Method	N2C	NB2NB	N2V	ZS-N2N	DIP	S2S	BM3D
GPU time	-	-	-	20 sec.	3 min.	35 min.	4 sec.
CPU time	-	-	-	80 sec.	45 min.	4.5 hr.	4 sec.
Network size	3.3M	1.3M	2.2M	22k	2.2M	1M	-

Table 4. Computational Resources. **First and Second Rows:** Time taken to denoise one image on average on GPU and CPU. The time for the dataset based methods is discarded, since it is negligible. BM3D does not benefit from the GPU, as there is no optimization involved. **Bottom Row:** Number of parameters of a network.

of noise model free techniques.

Regarding the three dataset free and noise model free methods considered, DIP was often lagging behind S2S and ZS-N2N, despite using the ground truths to find the best possible early stopping iteration. S2S's performance without ensembling is inadequate, and even with ensembling, it does not work well on low noise levels. Moreover, it requires more than 0.5 hours to denoise an image on a GPU and 4.5 hours on a CPU.

Except for ZS-N2N, all deep learning based baselines have millions of parameters, making them computationally expensive. Considering ZS-N2N's ability to generalize to various denoising conditions with relatively fast denoising time, very few parameters, and CPU compatibility, we can conclude that it offers a good trade-off between denoising quality and computational resources.

5. Conclusion

We proposed a novel zero-shot image denoising algorithm that does not require any training examples or knowledge of the noise model or level. Our work uses a simple 2-layer network, and allows denoising in a relatively short period of time even when executed without a GPU. The method can perform well on simulated noise as well as real-world camera and microscope noise, and achieves a good trade-off between generalization, denoising quality and computational resources compared to existing dataset free methods. A demo of our implementation including our code and hyperparameters can be found in the following colab notebook: https://colab.research.google.com/drive/1i82nyizTdszyHkaHBuKPbWnTzao8HF9b

Acknowledgements

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[2] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 524–533. PMLR, 2019. 2

[3] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing Images for Learned Raw Denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019. 2

[4] Sungmin Cha and Taesup Moon. Fully convolutional pixel adaptive image denoiser. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[5] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 477–485, 2015. 6

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021. 4

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1, 2, 5

[8] Mohammad Zalbagi Darestani and Reinhard Heckel. Accelerated mri with un-trained neural networks. *IEEE Transactions on Computational Imaging*, 7:724–733, 2021. 3, 6

[9] Mauricio Delbracio, Ignacio Garcia-Dorado, Sungjoon Choi, Damien Kelly, and Peyman Milanfar. Polyblur: Removing mild blur by polynomial reblurring. *IEEE Transactions on Computational Imaging*, 7:837–848, 2021. 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[11] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *International Conference on Learning Representations*, 2019. 3

[12] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14776–14785, 2021. 1, 2, 3, 5

[13] Sandip M. Kasar and Sachin D. Ruikar. Image demosaicking by nonlocal adaptive thresholding. In *2013 International Conference on Signal Processing , Image Processing Pattern Recognition*, pages 34–38, 2013. 5

[14] Chaewon Kim, Jaeho Lee, and Jinwoo Shin. Zero-shot blind image denoising via implicit neural representations, 2022. 2

[15] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void - learning denoising from single noisy images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2124–2132, 2019. 1, 2, 5

[16] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 5

[17] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2965–2974. PMLR, 2018. 1, 2, 3, 5

[18] Jason Lequyer, Reuben Philip, Amit Sharma, Wen-Hsin Hsu, and Laurence Pelletier. A fast blind zero-shot denoiser. *Nature Machine Intelligence*, oct 2022. 2, 6

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 2

[20] Markku Makitalo and Alessandro Foi. Optimal inversion of the anscombe transformation in low-count poisson image denoising. *IEEE Transactions on Image Processing*, 20(1):99–109, 2011. 1, 2, 7

[21] Youssef Mansour, Kang Lin, and Reinhard Heckel. Image-to-image mlp-mixer for image reconstruction. *CoRR*, abs/2202.02018, 2022. 2

[22] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 5

[23] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12061–12069, 2020. 2, 5, 7

[24] 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 (CVPR)*, 2020. 1, 2, 3, 5, 6

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors,

Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241, Cham, 2015. Springer International Publishing. 3, 4

[26] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

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

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 5

[29] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark, 2018. 6

[30] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[31] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: An open dataset and benchmarks for accelerated MRI. 2018. 6

[32] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2, 5

[33] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. A poisson-gaussian denoising dataset with real fluorescence microscopy images. In *CVPR*, 2019. 6

[34] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. Extending stein's unbiased risk estimator to train deep denoisers with correlated pairs of noisy images. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2, 5