Improved Noise2Noise Denoising with Limited Data

Adria Font Calvarons
Technical University of Munich
adria.font@tum.de

Abstract

Deep learning methods have proven to be very effective for the task of image denoising even when clean reference images are not available. In particular, Noise2Noise, which requires pairs of noisy images during the training phase, has been shown to yield results as good as approaches using pairs of noisy and clean images (Noise2Clean). However, the performance of Noise2Noise drops when the amount of training data is reduced, limiting its capability in practical scenarios.

In this work, an analysis of the Noise2Noise learning strategy is done using real noise and synthetic datasets. This paper demonstrates, using diverse network architectures and loss functions, that the duplicity of information in the noisy pairs can be exploited to reach increased denoising performance of Noise2Noise. Additionally, the issue of overfitting in Noise2Noise is analyzed, given its relevance when training with limited data, and an interpretable early termination criterion is proposed.

1. Introduction

Image denoising is a well studied problem in many applications. Traditionally, it has been posed as a restoration problem $\hat{x} = x + n$, where the variable of interest x is corrupted with additive white Gaussian noise (AWGN), resulting in the observed noisy image $\hat{x}$. This model has been used pervasively in denoising literature, yet several studies have highlighted the importance of focusing on realistic acquisition noise, given the several sources of noise present in most modern imaging systems [43, 28].
Early popular denoising approaches used nonlocal filtering [2, 6], and remarkably, they still serve as benchmark for recent developments in AWGN denoisng [23, 38, 21]. Other established methods rely on the low-rank of image features with respect to noise [45] or the possibility to represent images sparsely [44]. Other approaches attempt to embed popular denoisers as regularizer in minimization frameworks [5]. A more exhaustive discussion on denoising and restoration approaches is carried in [12], and [43].

Recent trends in image denoising indicate that deep learning approaches are superior to classical methods [1, 14]. These approaches, use pairs of noisy and clean images to estimate the optimal parameters of a convolutional neural network (CNN) [48, 51]. However, obtaining clean counterparts of images acquired from imaging devices is not trivial on most applications, including fluorescence microscopy [49], medical imaging [42] and even normal photography [29]. Additionally, there are applications in which obtaining clean images is not possible at all, for instance if the image's content is especially volatile (*e.g.* in cryo-electron microscopy [36, 10]).
For this reason, denoising methods that do not rely on clean data are of particular interest and several such deep learning based approaches have been proposed recently. With milestone method Noise2Noise, Lehtinen *et al.* [23] showed that pairs of noisy images of the same scenes can also be used, with equally good results. Building on this idea, other approaches require only single images instead of pairs during the training phase. Krull *et al.* [18] proposed a masking scheme to create blind spots in the receptive field of the network, preventing the regression problem to result in the network approximating an identity operator. The most recent variants of this idea introduce noise modelling to provide the algorithm with the information lost by the blind spot [19, 30, 20]. While this is beneficial for the denoising task, the usability of the method is conditioned by the availability and the quality of the noise model.
Altogether, when enough pairs of noisy images are available, the Noise2Noise approach [23] is a good candidate for denoising, it is straightforward, versatile, and does not impose major constraints on the noise characteristics (*e.g.* following a particular distribution), nor does it require much prior knowledge about the noise distribution. Additionally, it has found success in various applications [42, 23, 49, 15, 9, 3].

1.1. Motivation and Contribution

The Noise2Noise training method achieves the same performance as its traditional learning counterpart when

enough data is available [23]. Commonly used datasets for training denoising networks are designed for synthetic noise, which can be added on-the-fly, and are usually made up of several hundred to several thousand images [14, 38]. This also holds for the original Noise2Noise work [23], which reports that the performance of Noise2Noise learning is on-par with traditional learning. However, the smaller training datasets get, the less one can theoretically expect Noise2Noise to remain competitive [22]. Accordingly, other studies have shown a significant gap between networks trained with Noise2Noise and traditional strategies, usually when smaller sets have been used [18, 16, 42, 52].
The core of this paper is the proposal of a simple modification of the Noise2Noise training method that consistently enhances its denoising performance, getting closer to and often surpassing equivalent traditional learning methods, even when training on relatively small datasets. The experiments in this paper show that this holds across network architecture, loss function and noise characteristics.
As additional work, overfitting in Noise2Noise is studied. Generally, when training neural networks on small datasets, overfitting is a likely event. Therefore it is common to set a portion of the data aside (holdout) for validation and early termination purposes [35, 7, 24]. This paper also describes an interpretable Noise2Noise-specific termination criterion that does not require a holdout, together with initial observations on data.

2. Methods

2.1. Theoretical Background

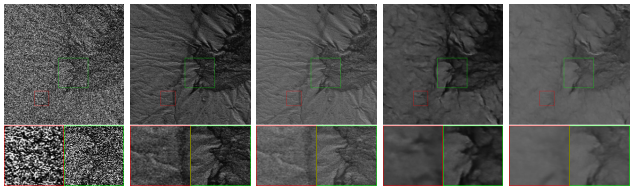
Traditional deep learning based methods [47, 14, 26] use k pairs of clean and noisy images $(x^i, \hat{x}^i)$ to estimate the optimal parameters θ of a network f_θ in a regression problem, which in the simplest cases has the following form:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{k} L(f_\theta(\hat{x}^i), x^i), \tag{1}$$

where L is a measure of dissimilarity between the network output and the clean target x. This approach is henceforth denoted Noise2Clean. A more practical alternative, Noise2Noise [23], replaces clean targets with noisy targets:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{k} L(f_\theta(\hat{x}^i), \hat{y}^i), \tag{2}$$

where, for a given pair i, $\hat{y}$ and $\hat{x}$ are now two independent noisy instances of the same scene. They may have a different noise level, as long as the expected value of the true images is the same, and the noise is zero mean.
In the original Noise2Noise work [23], several p-norms are used as the loss function L. There is a direct relationship between M-estimators such as the mean or the median and p-norms. For instance, it is trivial to show that regression using the ℓ^2 norm is equivalent to the mean of several observations, and that the same holds between the ℓ^1 norm and the median. A similar argument can be made for Eq. 2 since it also defines a regression problem. In an effort to minimize the distance between all the pairs of the problem, the minimization problem must find a balance between all the possible solutions of the problem, not unlike the computation of the mean or the median of several observations. This is illustrated in Fig. 1.



(a) noisy (b) median (c) mean (d) N2N$_{\ell^1}$ (e) N2N$_{\ell^2}$

Figure 1: The ℓ^1 and ℓ^2 norm loss functions as mean and median estimators. 99 noisy images similar to (a) have been aggregated using a per pixel median (b) and mean (c). (d) and (e) display the output of Noise2Noise networks trained with the ℓ^1 and ℓ^2 norms, respectively, which resemble the output of the corresponding M-estimator.

2.2. Maximizing Noise2Noise's use of limited data

When a Noise2Noise algorithm is trained with corrupted data on-the-fly, the network f_θ is given different noisy samples of input and target sets at each epoch. Providing multiple corrupted samples of the same latent image contributes positively to the learning task [23], since despite sharing the same underlying structure, the mapping from noisy image to noisy image is different, and the overall regression task is populated with more data, resulting in a better estimation of the true images (*e.g.* mean or median of possible solutions, for ℓ^1 and ℓ^2).
However, the assumption of unlimited noisy samples per scene is not realistic in many practical scenarios (*e.g.* if a noise model is not known). Let us assume only two noisy samples $(\hat{y}, \hat{x})$ per scene are available, the minimum amount that enables Noise2Noise learning. The goal now is to maximize the use of these two samples during the learning. Assuming noisy samples are drawn from the same distribution, one obvious idea is that they can both be used as input and target respectively (i.e. Eq. 2 can also have $\hat{y}$, $\hat{x}$ swapped). By doing so, the amount of pairs available for the Noise2Noise strategy is doubled, and one can expect gains in the resulting denoising performance. Henceforth, we refer to this approach as alternating Noise2Noise (AltN2N).
This idea can be taken one step further, by noting that,

in most cases, the regression problem in Eq. 2 computes element-wise differences of pixels but the result of a network f_θ at a given image coordinate depends on a larger region of pixels. The consequence of this is that changes in only a few pixels in the input or the target can yield virtually unseen samples of a scene.

In a Noise2Noise setting, the pixels (or pixel regions) in $\hat{y}$ and $\hat{x}$ are interchangeable as long as the images are well-aligned, and the noise is not correlated (or correlation is not destroyed in the process). Under this assumption, one can swap one or more single pixels (or pixel regions) between $\hat{y}$ and $\hat{x}$, such that two new unseen yet plausible images $\hat{y}_s$ and $\hat{x}_s$ are generated. These new images act as noise surrogates of the original pair, and several new pairs can be generated from the original pair, by randomly combining $\hat{y}$ and $\hat{x}$ into new disjoint images. Interpreting images with n pixels as points in an n-dimensional space, this swapping operation can be understood as sampling the remaining vertices of the n-dimensional hypercube that encloses all plausible samples given the information provided by $\hat{y}$ and $\hat{x}$. Although still in the vicinity of $\hat{y}$ and $\hat{x}$, all of these new points effectively populate the overall regression problem with more data. This is illustrated in Fig 2.

Lehtinen *et al*. [23] show that under a fixed capture budget (where clean images can be generated by averaging several noisy images) it is advantageous to use several noisy samples in a Noise2Noise scheme rather than generating a clean image out of several noisy samples in order to have a noisy-clean pair and use the traditional learning scheme. The generation of noise surrogates out of only two samples as capture budget, is intended to simulate the process of obtaining a large amount of noisy samples for a given scene.

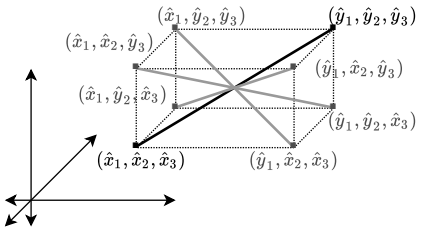
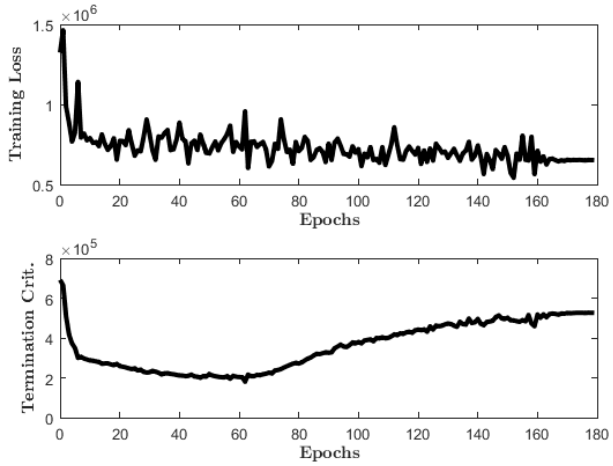
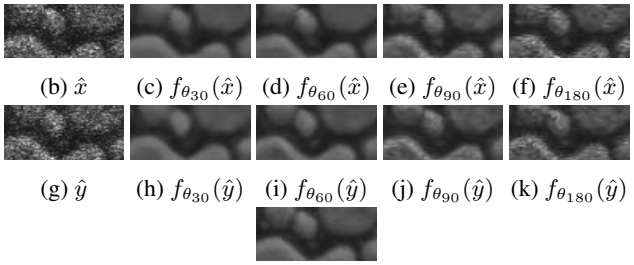


Figure 2: Illustration of the noise surrogates strategy, for a Noise2Noise image pair ($\hat{x}$,$\hat{y}$) of 3 pixels each.

A different motivation to use the noise surrogate technique, is to prevent overfitting, which may especially happen in small datasets. Figure 3 shows a minimal example of overfitting in Noise2Noise. Since the targets are noisy, the network eventually attempts to reproduce noise. In fact, due to that reason, Noise2Noise training might be more susceptible to overfitting than equivalent Noise2Clean training, especially with high levels of noise (i.e very low capture budget). With small but constant variation in input and target, reproducing the noise in the targets should become more challenging.



(a) training loss and proposed termination Criterion



(b) $\hat{x}$ (c) $f_{\theta_{30}}(\hat{x})$ (d) $f_{\theta_{60}}(\hat{x})$ (e) $f_{\theta_{90}}(\hat{x})$ (f) $f_{\theta_{180}}(\hat{x})$

(g) $\hat{y}$ (h) $f_{\theta_{30}}(\hat{y})$ (i) $f_{\theta_{60}}(\hat{y})$ (j) $f_{\theta_{90}}(\hat{y})$ (k) $f_{\theta_{180}}(\hat{y})$

(l) x

Figure 3: Illustration of overfitting in Noise2Noise and the proposed termination criterion, using a very small dataset. (a) Training Loss and Curve of the termination criterion on the training set. (b) and (f) are close-up images of one of the training pairs. (g) and (k) are corresponding intermediate outputs after a certain number of epochs (30,60,90,180) . (l) close-up of the corresponding clean image.

On a final practical note, this approach does not conflict with traditional augmentation methods. Additionally, in order to accommodate memory-performance requirements, noise surrogates can be generated on the fly in Eq. 2, once before every epoch, or a large number can also be precomputed before training. Henceforth a network trained with this technique is denoted surrogate Noise2Noise.

2.3. Overfitting in Noise2Noise and Early Termination

Overfitting (i.e. the network reaches states where it performs well the task on training data but performs poorly to unseen data) is an undesired effect to be monitored es-

pecially when training on small datasets. This is typically done by computing any meaningful measure (*e.g.* loss, accuracy) on the validation set, a separate set that does not participate in the minimization problem (hold-out). At some point during training, the network reaches an optimal state, after which it overfits. Discovering this inflection point is the goal of early termination [4, 31], and is a popular approach to prevent overfitting in denoising and other applications [35, 7, 24]. However, setting a portion of the training data aside for validation has several drawbacks, when data is limited.

First, the amount of data actively participating in the minimization problem is reduced by the partition. This especially compromises the performance of the Noise2Noise algorithm, with respect to an equivalent Noise2Clean algorithm [22].

Second, if the performance of the algorithm is evaluated on a too small validation set, it is likely to have a large stochastic error [24]. Additionally, it is not easy to guarantee that the validation set is representative of the training and test sets. Even when strategies such as k-fold cross validation are used, in the case of small datasets, the estimation of the performance by the evaluation set is especially sensitive to the way data is split into training and validation [46].

Third, in order to regularly obtain the performance estimates, the network needs to be applied regularly on the validation set. This computation results in a constant overhead to the algorithm's training time. Furthermore, when datasets are small, it is advisable to use time-consuming k-fold cross validation or similar strategies [11, 17].

All these problems could be solved if an estimate of the performance of the network during training could be made directly on the training set, and work in this direction has been done in the past [24, 8].

Noise2Noise is a special case, given that the targets of the regression problem are not actually the desired output images, but noisy images as well. Therefore, absolute overfitting can be said to happen whenever the network perfectly reproduces the noise in the targets of the training set, a trend which becomes obvious after a certain amount of epochs as shown in Fig 3. Assuming $\hat{x}$ and $\hat{y}$ are used both as input and target to one another, then a perfectly overfitted network is such that $f_\theta(\hat{x})$ approximates $\hat{y}$ and $f_\theta(\hat{y})$ approximates $\hat{x}$. In that case, the difference between the outputs $f_\theta(\hat{x})$ and $f_\theta(\hat{y})$ would be considerable. This means that a poor performance of the network can already be detected from the training set. On the other hand, if the network is truly efficient at denoising, the difference between $f_\theta(\hat{x})$ and $f_\theta(\hat{y})$ should be minimal, since the latent image is the same in both inputs, the only difference between them being the noise. From these observations the best performing network state θ will be such that

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{k} L(f_\theta(\hat{x}^i), f_\theta(\hat{y}^i)). \tag{3}$$

It is important to note that this cannot be a loss function, since it does not guarantee fidelity to the latent images (*e.g.* The problem in Eq. 3 has a clear minimum when the network forces all output pixels to be 0). However, when computed on the training set, besides the actual loss function, this measure would be an indicator of when the noise in the targets is attempted to be reproduced, and would, upon divergence, signal the need for training to be terminated, yielding an Early Termination rule.

This measure has a more clear interpretation than the validation loss in a Noise2Noise setting (since the reference images are noisy). On the other hand, one cannot exclude the possibility that previous to the overfitting defined by Eq. 3 the network is not overfitting to certain features of the training set, yet it does indicates a point after which overfitting is certain and happens due the network replicating the noisy targets. To illustrate these ideas with a minimal example, Fig. 3 shows how at some point during the training an optimal state is reached (according to a reference Fig. 3l), approximately at the moment where the outputs $f_\theta(\hat{x})$ and $f_\theta(\hat{y})$ look more similar (Figures 3i and 3d), which in turn coincides with the inflection point in Fig. 3a, representing Eq. 3.

In order for $f_\theta(\hat{x}^i)$ and $f_\theta(\hat{y}^i)$ to be available at every iteration, they both need to be computed, and could equally contribute to the regression problem:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{k} \frac{1}{2} L(f_\theta(\hat{x}^i), \hat{y}^i) + \frac{1}{2} L(f_\theta(\hat{y}^i), \hat{x}^i). \tag{4}$$

3. Experiments

The main goal of the experiments is to show that, under several different circumstances, the proposed Noise Surrogate technique contributes positively to the denoising performance of Noise2Noise. We start with various amounts of synthetic noise and continue the same experiments with real microscopy data. This is discussed in sections 3.1 and 3.2. Additionally, in section 3.2 we compare the curves of the termination criterion with those of the validation loss in different circumstances to reason about its usefulness.

3.1. Noise Surrogates and Synthetic Noise

In order to properly study the denoising capability of several methods it is important to have high-quality images as reference to quantify the fidelity of the restored images. Therefore, these reference images should be free of noise and distortion (*e.g.* compression artifacts). For this reason, we use the TAMPERE17 dataset [29], which quantifies and

guarantees excellent image quality for grayscale images. 200 images are allocated for training, 50 for validation and 50 for testing. Noise is added synthetically to the reference images in order to simulate noisy images. Additionally, after adding noise, the images are saved in 8-bit precision, which results in a certain amount of quantization error.

The Gaussian distribution at various standard deviation levels is a simple and popular noise generator in the denoising literature [37, 32]. Therefore, the proposed methods are tested with the gaussian standard deviation range $\sigma = (10, 30, 50)$.

The goal of experiments with synthetic noise is to show that the proposed method has an advantage over Noise2Noise generally across choice of loss function and network architecture, and to this end, various network architectures and loss functions are tested. First, the original U-Net Noise2Noise network architecture from [23] (described in detail in [22]) is adapted such that it does not add noise on-the-fly and the noise surrogate generation runs previous to every iteration. With that, the ℓ^2 norm is used as loss function, since it is appropriate for Gaussian noise [50]. In additional experiments, the more recent BRDNet network [38] architecture is also used as backbone. Likewise, the following loss function is also used (as an adaptation of the methods described in [39]):

$$L_{\text{H+MSSSIM}}(x, y) = (1 - \alpha)L_H(x, y) + \alpha MSSSIM(x, y), \tag{5}$$

where $\alpha \in (0, 1)$ is fixed to a constant. This loss is the combination of the Huber loss (as the Moreau envelope of non-differentiable ℓ^1 norm [25]) and MS-SSIM [41] metric.

For each combination of network architecture, loss function and noise level, the learning parameters are fixed such that the traditional Noise2Noise (N2N) method converges, and the same parameters are used for all the Noise2Noise variants described in section 2.2. As baseline, the same network is trained with clean targets (Noise2Clean). For all methods, the network state achieving lowest validation loss throughout the training is saved and tested on the test set using popular PSNR (Peak Signal-to-Noise Ratio) and SSIM [40] quality metrics.

The results for each noise level are summarized in Table 1 and example image results are displayed in Fig. 5. That providing more noisy examples of the same scenes to the Noise2Noise learning is advantageous can be clearly seen in the table. There is clear indication that noise surrogate generation provides new plausible noisy samples, since the surrogate Noise2Noise (SN2N) consistently performs better than the alternating Noise2Noise (AltN2N), which in turn outperforms the traditional Noise2Noise (N2N). Noticeably, when the noise magnitude is small the surrogate technique can improve the performance of Noise2Noise up to or above the levels of Noise2Clean (N2C). Additionally, Fig. 4 shows the trends of different methods. It can be seen that while the Noise2Clean strategy quickly reaches higher overall metric values, using the noise surrogate technique eventually causes the learning trend to dissociate from the trends of the other Noise2Noise techniques and eventually reach levels comparable to those of the Noise2Clean approach.

The performance improvement is evident, yet the proposed surrogate Noise2Noise inherits some of the limitations of the traditional Noise2Noise method. The third row in Fig. 5 reveals the limitation of the noise distribution being assumed 0 mean in all Noise2Noise approaches. Because the noise deviates from perfect 0 mean in very dark or bright areas of the image (due to clipping and quantization in the range 0 - 255), there is a generalized mean error for all Noise2Noise approaches in the darker areas.

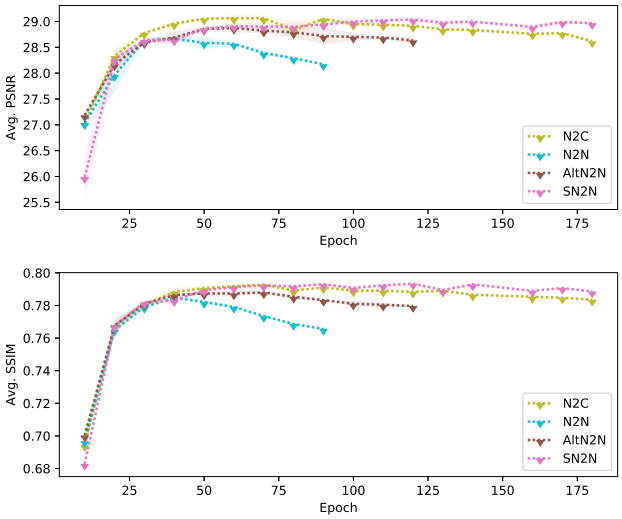


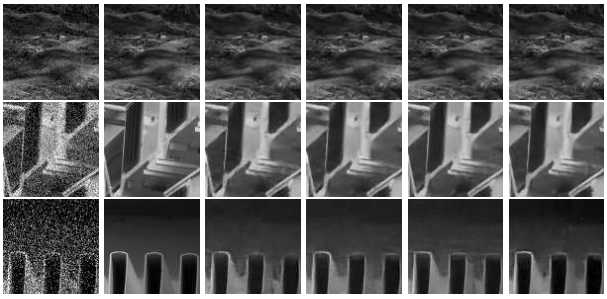
Figure 4: Average trends of PSNR(dB) and SSIM along training among 5 training runs. Networks using different learning methods have been saved at constant intervals and used to compute PSNR and SSIM. Training has been interrupted when overfitting acording to the validation set was evident. For this illustration, all methods use the ℓ^2 norm and the U-Net network with same parameters on the TAMPERE17 dataset with $\sigma = 30$.

3.2. Noise Surrogates and Real Data

Three in-house electron microscopy datasets of different specimens acquired using different acquisition parameters have been used for evaluating the proposed strategy. Example images are shown in Fig 6. For each scene of each dataset, 99 images have been obtained. Two of these images are saved as $\hat{x}$ and $\hat{y}$. Both the mean and median aggrega-

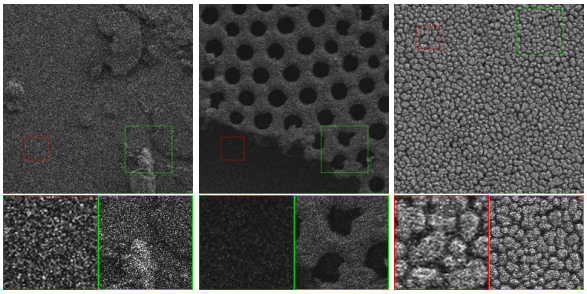
	$\sigma = 10$		$\sigma = 30$		$\sigma = 50$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
U-Net N2C$_{\ell^2}$	**34.5874**	**0.9209**	**29.1075**	**0.7922**	**26.9587**	**0.7162**
U-Net N2N$_{\ell^2}$	34.3260	0.9178	28.6850	0.7845	26.0085	0.6980
U-Net AltN2N$_{\ell^2}$	34.3605	0.9202	28.8865	0.7875	26.0851	0.7039
U-Net SN2N$_{\ell^2}$	**34.5876**	**0.9213**	29.0544	**0.7918**	26.3837	0.7137
U-Net N2C$_{L_{\text{H+MSSSIM}}}$	34.5080	0.9201	**29.1305**	**0.7972**	**26.9581**	**0.7197**
U-Net N2N$_{L_{\text{H+MSSSIM}}}$	34.0468	0.9182	28.4631	0.7798	25.7911	0.6915
U-Net AltN2N$_{L_{\text{H+MSSSIM}}}$	34.2641	0.9196	28.5562	0.7875	26.1675	0.7038
U-Net SN2N$_{L_{\text{H+MSSSIM}}}$	**34.5135**	**0.9214**	28.9889	**0.7968**	26.2981	0.7133
BRD-Net N2C$_{\ell^2}$	**34.6708**	**0.9203**	**29.2460**	0.7931	**27.0160**	0.7096
BRD-Net N2N$_{\ell^2}$	34.3689	0.9174	28.6932	0.7780	25.9583	0.6804
BRD-Net AltN2N$_{\ell^2}$	34.5113	0.9200	28.8857	0.7852	26.0827	0.6944
BRD-Net SN2N$_{\ell^2}$	34.5767	**0.9208**	29.1521	**0.7948**	26.4197	**0.7134**

Table 1: Results, PSNR(dB) and SSIM, for the TAMPERE17 dataset, including the different Noise2Noise variants and the Noise2Clean counterpart on different combinations of loss function, network and noise level. The best results in each category are highlighted.



(a) noisy (b) clean (c) N2N (d) AltN2N (e) SN2N (f) N2C

Figure 5: Example close-up results for the TAMPERE17 dataset on the U-Net Architecture, the ℓ^2 norm loss function and the several training versions described. Top using $\sigma = 10$. Middle using $\sigma = 30$. Bottom using $\sigma = 50$.



(a) Dataset 1 (b) Dataset 2 (c) Dataset 3

Figure 6: Sample images belonging to the in-house microscopy dataset.

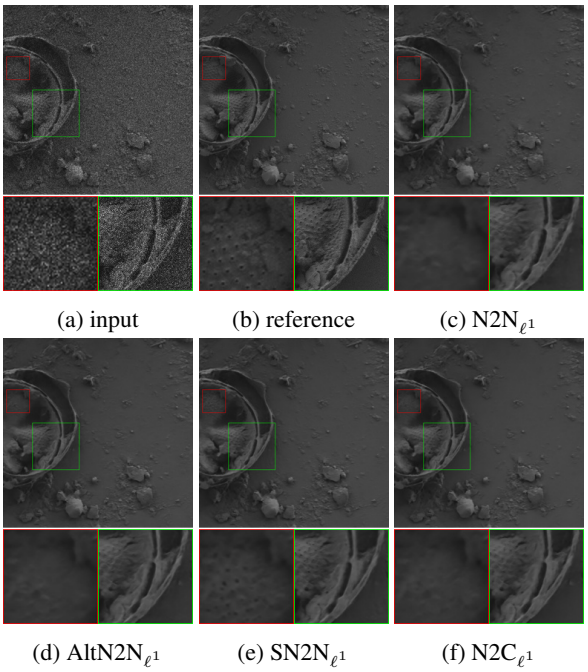
tion of the 99 images have been computed, and are used as ground truth with the respective loss function (the ℓ^1 and ℓ^2 norm loss functions). Alignment is computed in a sub-pixel exact way using the cross-correlation method described by Guizar-Sicairos *et al*. [13]. Misaligned scene sets have been discarded. After this preprocessing, datasets 1 and 2 have 84 1024×768 scenes for training, 15 512×768 scenes for validation and 15 512×512 scenes for testing. Dataset 3 has 38 1024×768 scenes for training, 14 512×768 scenes for validation and 14 512×512 scenes for testing. For the training and validation sets, 256×384 regions have been cropped from the images in order to be fed to the networks. Additionally, simple data augmentation involving flipping (left-to-right, up-to-down and both) is used.

The same Noise2Noise variants and setup are used as described in section 3.1, with the only difference that noise surrogates are generated by swapping entire horizontal lines instead of single pixels. The reason for that is that electron microscopy images contain horizontal noise correlation [33, 34]. Therefore, instead of swapping single pixels, swapping entire horizontal lines is the correct noise surrogate generation approach, in order not to artificially modify the noise characteristics that the network is expected to cope with once trained.

Metric results of the different approaches are summarized in table 2. Similar to the results for synthetic noise, one can observe that the SN2N strategy enhances the performance of the original Noise2Noise up to the level of its Noise2Clean counterpart or even surpassing its performance. Example result images are shown in Fig. 7.

	Dataset-1		Dataset-2		Dataset-3	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
U-Net N2C$_{\ell^1}$	**28.3522**	0.5585	31.7210	**0.7509**	27.4497	0.8401
U-Net N2N$_{\ell^1}$	28.1437	0.5570	31.6518	0.7465	27.3359	0.8328
U-Net AltN2N$_{\ell^1}$	28.2461	0.5555	31.7324	0.7479	27.4755	0.8380
U-Net SN2N$_{\ell^1}$	28.3436	**0.5589**	**31.7665**	0.7497	**27.6054**	**0.8440**
U-Net N2C$_{\ell^2}$	**29.1264**	**0.6122**	32.1382	0.7736	27.6987	0.8511
U-Net N2N$_{\ell^2}$	28.9760	0.6045	32.0096	0.7690	27.5145	0.8430
U-Net AltN2N$_{\ell^2}$	28.9648	0.6005	32.0975	0.7698	27.7186	0.8504
U-Net SN2N$_{\ell^2}$	29.1222	0.6097	**32.1720**	**0.7751**	**27.8383**	**0.8561**

Table 2: Results, PSNR(dB) and SSIM, of the different Noise2Noise variants and the Noise2Clean. The results of algorithms using the ℓ^2 and ℓ^1 norms have been compared against the mean and median of 99 aligned noisy images, respectively. The targets of the N2C are chosen accordingly.



(a) input (b) reference (c) N2N$_{\ell^1}$

(d) AltN2N$_{\ell^1}$ (e) SN2N$_{\ell^1}$ (f) N2C$_{\ell^1}$

Figure 7: Sample results for the different methods on images belonging to the microscopy dataset.

3.3. Termination criterion and validation loss

For all results in this section, standard Noise2Noise training with the ℓ^2 norm loss function is performed. In parallel, the termination criterion (Eq. 3) is computed on the training and validation sets separately using the same norm. The setup is such that it surely leads to strong overfitting, and the denoising performance is not relevant. In order to show that the proposed termination criterion can correctly estimate overfitting, the following observations are experimentally shown:

First, with enough training iterations the termination criterion computed on the training set eventually diverges. Additionally, the termination criterion reacts to learning rate changes similarly to the validation loss.
Second, when the termination criterion is computed in the validation set during training, it starts to diverge in accordance with divergence as observed by the validation loss.
Third, if the training set and the validation set are perfectly representative of one another, then the termination criterion computed on the training set correlates well with the termination criterion computed on the test set and, by extension, also with the validation loss. Therefore, the termination criterion computed on the training set is indicative of when to halt the training.
For the first two observations, we use the datasets, with the same partition as described in previous sections. We observe there is generally an inflection point after which divergence is apparent, additionally, when the magnitude of the learning rate is brought down, the termination criterion on the training set indicates stronger overfitting, just like the validation loss does. This is exemplary shown in Figs. 9a and 9b. In these figures it can also be seen that the validation loss and the termination criterion computed on the validation set start to diverge at the same epoch, albeit the former is more noisy.
Showing that the termination criterion computed on the training set has a similar trend than that of the one computed on the validation set is conditioned to the training and validation sets being representative of one another. This property is hard to guarantee when partitioning small and diverse image datasets (such as TAMPERE17). For this reason, a small dataset of simple synthetic images based on Perlin Patterns [27] is generated, with synthetically added Gaussian noise ($\sigma = 50$). The dataset is made up of 20 image pairs for training and 20 image pairs for validation, and each image is 256×256 pixels. Example images are

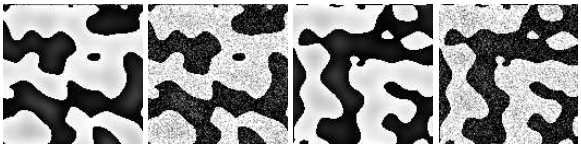


Figure 8: Example images (close-up) of Perlin Noise Patterns, with the noisy versions used in the study and the corresponding latent images.

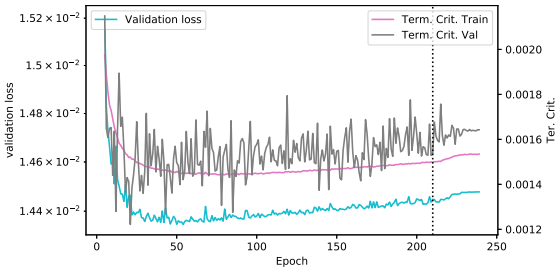
shown in Fig 8. The patterns are random but can be said to be drawn from the exact same distribution for training and validation. This is the reason why the termination criterion computed in the training and validation sets appears to be much more correlated (Figs. 9c and 9d). Furthermore, both of these curves start to diverge similarly to the validation loss.

These observations indicate that the termination criterion is relatively orthogonal to the loss function, and therefore, the proposed interpretable measure derived from the training set alone can be indicative of overfitting. However, while the trends during learning are illustrative, we acknowledge the need to derive a quantitative measure in future work. Another limitation of these observations is that they hold in situations with high levels of noise, where overfitting to noise is more likely to occurr early during training.

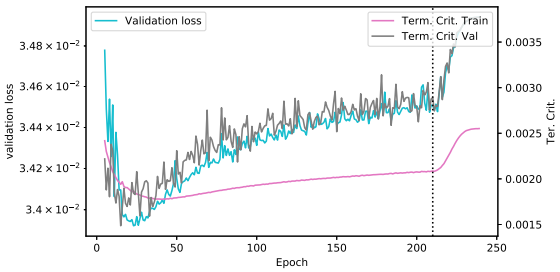
4. Conclusion

The main contribution of this paper is to show that swapping pixels (or pixel regions) between the images in the training data pairs allows to enhance the performance of Noise2Noise generally. This is achieved while not imposing any further requirement on the data, other than handling of noise correlation. The proposed idea is extremely simple to implement, causes a negligible overhead on top of the original algorithm, and can be especially relevant in cases where limited data is available (*e.g.* biomedical data). Additionally, the issue of overfitting in Noise2Noise is analyzed. A set of observations is collected indicating that the difference between the denoised images in the training pairs can be an estimator of the performance of the Noise2Noise network and can be used as a termination criterion during training. Both these contributions are expected to be of high interest for the several applications where Noise2Noise has been shown to be useful.

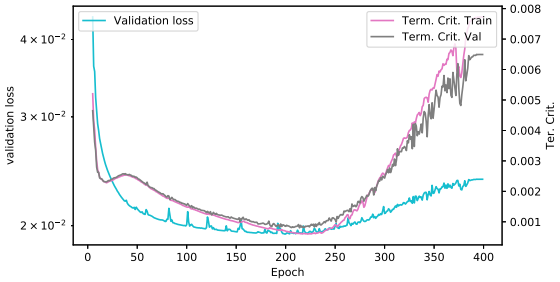
This study has been centered around grayscale images, driven by the electron microscopy domain data. As future work, authors intend to study how well these ideas translate to color images, especially considering the noise surrogation technique with the color channel dimension. In further study, authors also intend to quantify the ideas surrounding the proposed termination criterion on bigger datasets.



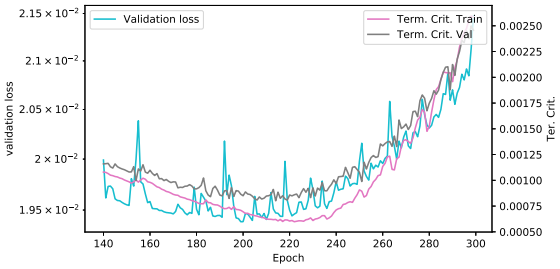
(a) N2N$_{\ell^2}$, TAMPERE17, $\sigma = 30$



(b) N2N$_{\ell^2}$, TAMPERE17, $\sigma = 50$



(c) N2N$_{\ell^2}$, Perlin Pattern, $\sigma = 50$



(d) N2N$_{\ell^2}$, Perlin Pattern, $\sigma = 50$ (close up)

Figure 9: Curves of the termination criterion and the validation loss on different datasets. The cyan curve is the validation loss and the pink and gray curves are training and validation versions of the termination criterion, respectively. In 9a and 9b the vertical dashed line indicates the point after which the learning rate has been reduced towards 0.

References

[1] Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. Ntire 2019 challenge on real image denoising: Meth-

ods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[2] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[3] Tim-Oliver Buchholz, Mareike Jordan, Gaia Pigino, and Florian Jug. Cryo-care: content-aware image restoration for cryo-transmission electron microscopy data. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 502–506. IEEE, 2019.

[4] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408, 2001.

[5] Stanley H Chan. Performance analysis of plug-and-play admm: A graph signal processing perspective. *IEEE Transactions on Computational Imaging*, 5(2):274–286, 2019.

[6] 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.

[7] Alexander de Brebisson and Giovanni Montana. Deep neural networks for anatomical brain segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 20–28, 2015.

[8] David Duvenaud, Dougal Maclaurin, and Ryan Adams. Early stopping as nonparametric variational inference. In *Artificial Intelligence and Statistics*, pages 1070–1077, 2016.

[9] Thibaud Ehret, Axel Davy, Jean-Michel Morel, Gabriele Facciolo, and Pablo Arias. Model-blind video denoising via frame-to-frame training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11369–11378, 2019.

[10] Stanley L Erlandsen, Cecile Ottenwaelter, Chris Frethem, and Ya Chen. Cryo field emission scanning electron microscopy. *BioTechniques*, 31(2):300–305, 2001.

[11] Yves Grandvalet and Yoshua Bengio. Hypothesis testing for cross-validation. *Montreal Universite de Montreal, Operationnelle DdIeR*, 1285, 2006.

[12] Shuhang Gu and Radu Timofte. A brief review of image denoising algorithms and beyond. *Springer series on Challenges in Machine Learning*, 1, 2019.

[13] Manuel Guizar-Sicairos, Samuel T Thurman, and James R Fienup. Efficient subpixel image registration algorithms. *Optics letters*, 33(2):156–158, 2008.

[14] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[15] Sai Gokul Hariharan, Christian Kaethner, Norbert Strobel, Markus Kowarschik, Shadi Albarqouni, Rebecca Fahrig, and Nassir Navab. Learning-based x-ray image denoising utilizing model-based image simulations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 549–557. Springer, 2019.

[16] Saeed Izadi, Zahra Mirikharaji, Mengliu Zhao, and Ghassan Hamarneh. Whitenner-blind image denoising via noise whiteness priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[17] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[18] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.

[19] Alexander Krull, Tomas Vicar, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *arXiv preprint arXiv:1906.00651*, 2019.

[20] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems*, pages 6968–6978, 2019.

[21] Stamatios Lefkimmiatis. Universal denoising networks: a novel cnn architecture for image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3204–3213, 2018.

[22] Jaakko Lehtinen, Jacob Munkberg1 Jon Hasselgren1 Samuli Laine, and Tero Karras1 Miika Aittala3 Timo Aila. Supplemental material (noise2noise).

[23] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, Timo Aila, et al. Noise2noise. In *International Conference on Machine Learning*. PMLR, 2018.

[24] Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*, 2017.

[25] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[26] Bumjun Park, Songhyun Yu, and Jechang Jeong. Densely connected hierarchical network for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[27] Ken Perlin. Improving noise. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 681–682, 2002.

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

[29] Mykola Ponomarenko, Nikolay Gapon, Viacheslav Voronin, and Karen Egiazarian. Blind estimation of white gaussian noise variance in highly textured images. *Electronic Imaging*, 2018(13):382–1, 2018.

[30] Mangal Prakash, Manan Lalit, Pavel Tomancak, Alexander Krull, and Florian Jug. Fully unsupervised probabilistic noise2void. *arXiv preprint arXiv:1911.12291*, 2019.

[31] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[32] Tal Remez, Or Litany, Raja Giryes, and Alex M Bronstein. Class-aware fully convolutional gaussian and pois-

son denoising. *IEEE Transactions on Image Processing*, 27(11):5707–5722, 2018.

[33] Joris Roels, Jan Aelterman, Jonas De Vylder, Hiep Luong, Yvan Saeys, Saskia Lippens, and Wilfried Philips. Noise analysis and removal in 3d electron microscopy. In *International Symposium on Visual Computing*, pages 31–40. Springer, 2014.

[34] Joris Roels, Jan Aelterman, Jonas De Vylder, Hiep Luong, Yvan Saeys, and Wilfried Philips. Bayesian deconvolution of scanning electron microscopy images using point-spread function estimation and non-local regularization. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 443–447. Ieee, 2016.

[35] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In *Advances in Neural Information Processing Systems*, pages 3257–3267, 2018.

[36] Robert E Thach and Sigrid S Thach. Damage to biological samples caused by the electron beam during electron microscopy. *Biophysical journal*, 11(2):204–210, 1971.

[37] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *arXiv preprint arXiv:1912.13171*, 2019.

[38] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020.

[39] AFM Shahab Uddin, Taechoong Chung, and Sung-Ho Bae. A perceptually inspired new blind image denoising method using $l\_\{1\}$ and perceptual loss. *IEEE Access*, 7:90538–90549, 2019.

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[41] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[42] Dufan Wu, Kuang Gong, Kyungsang Kim, Xiang Li, and Quanzheng Li. Consensus neural network for medical imaging denoising with only noisy training samples. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 741–749. Springer, 2019.

[43] 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.

[44] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. *ECCV*, 2018.

[45] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1096–1104, 2017.

[46] Yun Xu and Royston Goodacre. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, 2(3):249–262, 2018.

[47] 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.

[48] 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.

[49] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11710–11718, 2019.

[50] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.

[51] Yuqian Zhou, Jianbo Jiao, Haibin Huang, Yang Wang, Jue Wang, Honghui Shi, and Thomas Huang. When awgn-based denoiser meets real noises. *arXiv preprint arXiv:1904.03485*, 2019.

[52] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. Extending stein's unbiased risk estimator to train deep denoisers with correlated pairs of noisy images. *arXiv preprint arXiv:1902.02452*, 2019.