# Self2Self With Dropout: Learning Self-Supervised Denoising From Single Image

Yuhui Quan[1], Mingqin Chen[1], Tongyao Pang[2] and Hui Ji[2]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2]Department of Mathematics, National University of Singapore, 119076, Singapore
csyhquan@scut.edu.cn, csmingqinchen@mail.scut.edu.cn, matpt@nus.edu.sg and matjh@nus.edu.sg

## Abstract

*In last few years, supervised deep learning has emerged as one powerful tool for image denoising, which trains a denoising network over an external dataset of noisy/clean image pairs. However, the requirement on a high-quality training dataset limits the broad applicability of the denoising networks. Recently, there have been a few works that allow training a denoising network on the set of external noisy images only. Taking one step further, this paper proposes a self-supervised learning method which only uses the input noisy image itself for training. In the proposed method, the network is trained with dropout on the pairs of Bernoulli-sampled instances of the input image, and the result is estimated by averaging the predictions generated from multiple instances of the trained model with dropout. The experiments show that the proposed method not only significantly outperforms existing single-image learning or non-learning methods, but also is competitive to the denoising networks trained on external datasets.*

## 1. Introduction

Image denoising is the process to remove measurement noises from noisy images. It not only has great practical value, but also serves as a core module in many image recovery tasks. A noisy image $\boldsymbol{y}$ is usually modeled as

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

where $\boldsymbol{x}$ denotes the clean image (ground truth), and $\boldsymbol{n}$ denotes the measurement noise often assumed to be random.

In recent years, deep learning has become a prominent approach for image denoising, which uses a set of training samples to train a deep neural network (NN), denoted by $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$ with the parameter vector $\boldsymbol{\theta}$, that maps a noisy image to its clean counterpart. Most existing deep-learning-based denoising methods (*e.g.* [26, 31, 32]) use many pairs of clean/noisy images, denoted by $\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_i$, as the training samples, and the training is done by solving

$$\min_{\boldsymbol{\theta}} \sum_i \mathcal{L}(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), \boldsymbol{y}^{(i)}), \tag{2}$$

where $\mathcal{L}(\cdot,\cdot)$ measures the distance between two images. The availability of a large number of training samples is one key factor contributing to the performance of these methods. Sometimes, it can be expensive and difficult to collect a large dataset of useful clean/noisy image pairs.

Recently, there are some studies on training denoising NNs with only external noisy images. The Noise2Noise (N2N) method [19] showed that a denoising NN model can be trained using many pairs of two noisy images of the same scene. Using a self-prediction loss, together with a so-called *blind-spot* strategy to avoid learning an identity mapping, the Noise2Void (N2V) method [15] and the Noise2Self (N2S) method [3] showed the possibility to learn a denoising NN with good performance on a set of unorganized external noisy images. Yet, to achieve good performance, the external images used for training should be highly related to the noisy image being processed, in terms of image content and noise statistics. The collection of such external images can be costly or challenging in practice.

It is of great value to develop a powerful denoising NN that has no prerequisite on training samples. That is, the denoising NN is learned only on the input image itself. So far, there has been very little work along this line. Based on the *deep image prior* (DIP), Ulyanov *et al*. [25] proposed a single-image deep learning model for image recovery. The aforementioned dataset-based N2V and N2S methods can also be trained using only a noisy image. However, the performance of these methods is not competitive to existing non-local methods, *e.g*. BM3D [10]. To summarize, there is no satisfactory solution on how to train a denoising NN with good performance, given only the input noisy image.

### 1.1. Aim and Basic Idea

Motivated by its practical value and the lack of good solutions, this paper aims at developing an NN-based denoiser, which has good performance and yet can be trained

on only the given noisy image. In other words, this paper studies how to train a denoising NN

$$\mathcal{F}_{\boldsymbol{\theta}}(\cdot): \quad \boldsymbol{y} \to \boldsymbol{x}, \tag{3}$$

using only the input noisy image $\boldsymbol{y}$ itself.

Compared to supervised deep learning, the single-image-based self-supervised learning is much more challenging. The over-fitting is much more severe when training an NN on a single image. A denoising NN can be interpreted as a Bayes estimator with its prediction accuracy measured by the mean squared error (MSE):

$$\text{MSE} = \text{bias}^2 + \text{variance}, \tag{4}$$

The variance will dramatically increase when the number of training samples decreases from many to one. The blind-spot technique [15, 3] can overcome one phenomenon of overfitting, *i.e*. the model converges to an identity mapping. However, it is not effective on reducing the large variance caused by a single training sample. As a result, existing blind-spot-based NNs, *e.g*. N2V and N2S, do not perform well when being trained on a single image. In short, variance reduction is the key for the self-supervised learning on a single image.

To reduce the variance of an NN-based Bayes estimator, our solution is the dropout-based ensemble. *Dropout* [24] is a widely-used regularization technique for deep NNs. It refers to randomly dropping out nodes when training an NN, which can be viewed as using a single NN to approximate a large number of different NNs. In other words, dropout provides a computationally-efficient way to train and maintain multiple NN models for prediction. Owing to model uncertainty introduced by dropout [12], the predictions from these models are likely to have certain degree of statistical independence, and thus the average of these predictions will reduce the variance of the result.

Indeed, dropout is closely related to the blind-spot strategy used in N2V for avoiding the convergence to an identity mapping. Note that the blind-spot strategy synthesizes multiple noisy versions of the noisy image $\boldsymbol{y}$ by randomly sampling $\boldsymbol{y}$ with replacement, and the loss for training is measured on those replaced samples. Thus, it can be viewed as some form of dropout in the first and the last layer of the NN with specific connectivity.

Based on the discussion above, we propose a dropout-based scheme for the single-image self-supervised learning of denoising NNs. Our scheme uses a self-prediction loss defined on the pairs of Bernoulli sampled instances of the input image. A Bernoulli sampled instance $\widehat{\boldsymbol{y}}$ of an image $\boldsymbol{y}$ with probability $p$ is defined by

$$\widehat{\boldsymbol{y}}[k] = \begin{cases} \boldsymbol{y}[k], & \text{with probability } p; \\ 0, & \text{with probability } 1-p. \end{cases}$$

Consider two sets $\{\widehat{\boldsymbol{y}}_m\}_m, \{\widetilde{\boldsymbol{y}}_n\}_n$ of independent Bernoulli sampled instances of $\boldsymbol{y}$. Two main components of the proposed scheme are outlined as follows.

- **Training**. Train the NN by minimizing the following loss function with Bernoulli dropout:

$$\min_{\boldsymbol{\theta}} \sum_m \mathcal{L}(\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}_m), \boldsymbol{y} - \widehat{\boldsymbol{y}}_m).$$

- **Test**. Feed each $\widetilde{\boldsymbol{y}}_n$ to the trained model with Bernoulli dropout to generate a prediction $\widetilde{\boldsymbol{x}}_n$. Then output the average of all predictions $\{\widetilde{\boldsymbol{x}}_n\}_n$ as the result.

**Remark 1.** Dropout is often seen when training a classification NN. Most NNs for image recovery are trained without dropout. Also, it is very rare to see the usage of dropout during test in image recovery. This paper shows that using dropout in both training and test is very effective on boosting the performance when training a denoising NN on only an input noisy image. The main reason is that it can effectively reduce the variance of the prediction.

### 1.2. Contributions and Significance

In this paper, we present a self-supervised dropout NN, called Self2Self (S2S), for image denoising, which allows being trained on a single noisy image. See the following for the summary of our technical contributions.

- **Training a denoising NN using Bernoulli sampled instances, with a partial-convolution-based implementation**. Given only a noisy image without ground truth, we propose to use its Bernoulli sampled instances for training the NN with mathematical justification. Also, the partial convolution is used to replace the standard one for re-normalization on sampled pixels, which further improves the performance.
- **Using Bernoulli dropout in both training and test for variance reduction**. Interpreting a denoising NN as a Bayes estimator, the variance reduction is the key for single-image self-supervised training. Built upon the model uncertainty introduced by dropout, we propose to use Bernoulli dropout in both the training and test stages for reducing the variance of the prediction.
- **Solid performance improvement over existing solutions.** Extensive experiments on blind denoising under different scenarios show that the proposed approach outperforms existing single-image methods by a large margin. More importantly, its performance is even competitive to the denoising NNs trained on external image datasets, *e.g*. N2N.

The work in this paper has significance for both research and applications. The deep denoising NN has been a very basic tool in recent development of image recovery methods. However, most existing NN-based methods have the

prerequisite on a large amount of training data relevant to the target images, which limits their broader applicability. The issue on data collection remains, even though some methods only need noisy/noisy image pairs (*e.g*. N2N) or unorganized noisy images (*e.g*. N2V, N2S). An image denoising NN without prerequisite on training data is very welcomed in practice owing to its convenience.

Despite the importance of single-image self-supervised learning for image denoising NNs, there are few solutions and their performance is not competitive to those dataset-based learning methods. This paper shows that it is possible to train a denoising NN with competitive performance, using a single noisy image itself. The results presented in this paper on self-supervised learning for single image not only provide an NN-based image denoiser that is attractive in practice, but also can inspire further investigations on self-supervised learning to other image restoration problems.

## 2. Related Work

There is abundant literature on image denoising. The following review is focused more on the learning-based approaches closely related to our work.

**Non-learning based image denoisers.** A large number of image denoisers are non-learning-based and they impose some pre-defined image priors on the ground truth image to guide the denoising. One widely-used prior in image denoising is the sparsity prior of image gradients, which leads to various $\ell_p$-norm relating regularization methods, *e.g*. total variation denoising [6]. Another prominent one is the patch recurrence prior employed by the non-local methods. Among those, BM3D [10] is one of the top performers, which applies collaborative filtering to similar patches.

**Image denoisers learned on clean/noisy image pairs.** In recent years, many supervised learning methods are developed for image denoising, which learn the denoiser on a set of clean/noisy image pairs. Some of them learn the parameters of unfolded denoising processes; *e.g*. [23, 9, 30]. The more prominent ones train deep NNs as the denoisers; see *e.g*. [26, 31, 32, 8, 18, 13, 14]. Among them, the DnCNN [31] that uses residual learning for blind denoising is a common benchmark for NN-based image denoisers.

**Deep image denoisers trained with multiple noisy images**. Instead of using the pairs of clean/noisy images for training, the aforementioned N2N method [19] successfully trains a denoising NN using the pairs of two noisy images of the same scene. Its performance is close to that of NNs trained using clean/noisy pairs. Indeed, as long as the noise of the noisy/noisy pair is independent, the expectation of MSE of such a pair is the same as that of the clear/noisy pair. Nevertheless, the collection of many image pairs can still be difficult. Cha *et al*. [5] alleviated this problem by synthesizing noisy image pairs using GAN.

Instead of using organized noisy image pairs, some approaches [15, 16, 3, 17] use only unorganized noisy images for NN training, which is done by defining an effective self-prediction loss. Given a set of noisy images $\{\boldsymbol{y}_i\}_i$, training the NN using the standard loss function, $\sum_i \mathcal{L}(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}_i), \boldsymbol{y}_i)$, can lead to severe overfitting such that $\mathcal{F}_{\boldsymbol{\theta}}$ converges to an identity mapping. Avoiding the convergence to an identity mapping has been one main concern of self-supervised learning in image denoising.

The auto-encoder-based denoising NN [27] addresses such a concern using the architecture which excludes identity mappings, yet its performance is unsatisfactory. The blind-spot mechanism proposed in N2V [15] avoids learning an identity mapping by only allowing the NN to predict each pixel by its neighboring pixels. The implementation is done by randomly choosing image pixels of a noisy image and replacing the value of each chosen pixel by the value of a randomly-chosen neighboring pixel, and the loss is only computed on the image pixels with replaced values. Similar schemes are used in a parallel work N2S [3] and N2V's probabilistic extension [16]. Laine *et al*. [17] built the blind-spot mechanism into its NN architecture by excluding the center pixel in its receptive field.

**Image denoisers learned from only a single noisy image.** A learning-based image denoiser without any prerequisite on training samples is the most flexible to employ in practice. The sparse-coding-based denoisers learn a dictionary [11, 1, 2, 22] or a wavelet tight frame [4] from the noisy image, and the denoising result is defined as a sparse approximation to the input over the learned system.

There are few studies on training denoising NNs using only one single noisy image. One is the DIP method [25]. It assumes that, when learning an NN to approximate a degraded image, meaningful image patterns are learned with the priority over random patterns such as noise. Thus, it trains a generative NN that maps a random input to the given degraded image, which is regularized by early stopping. Despite its simplicity, the performance of DIP is not satisfactory and may be sensitive to the iteration number whose optimal value is hard to determine. By defining the training data as only a single noisy image, the aforementioned N2V and N2S can be extended to the case of single-image learning. However, their performance is not competitive either.

## 3. Main Body

This section starts with the introduction of the architecture of our Self2Self NN, followed by a detailed discussion on the schemes for self-supervised training and denoising.

### 3.1. NN Architecture

The diagram of the proposed Sefl2Self NN is shown in Fig. 1. Briefly, it is an encoder-decoder NN. Given an input

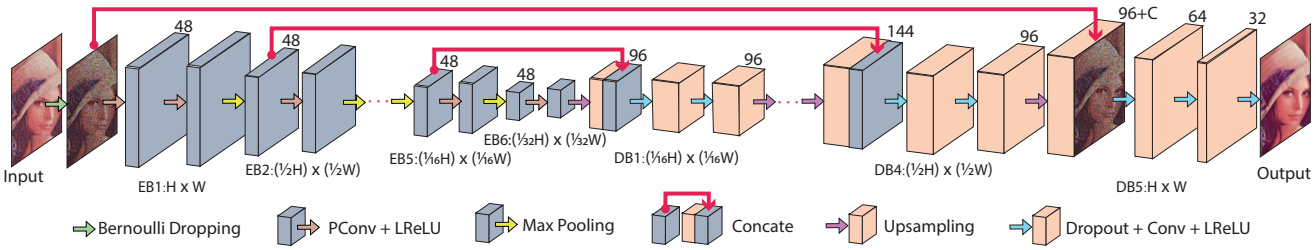


Figure 1. Architecture of proposed Self2Self NN.

noisy image of the size $H \times W \times C$, the encoder first maps the image to an $H \times W \times 48$ feature cube with a partial convolutional (PConv) layer [20], which is then processed by the following six encoder's blocks (EBs). Each of the first five EBs sequentially connects a PConv layer, a leaky rectified linear unit (LReLU), and a max pooling layer with $2 \times 2$ receptive fields and with a stride of 2. The last EB contains a PConv layer and an LReLU. The number of channels is fixed to 48 across all EBs. The output of the encoder is then a feature cube of size $H/32 \times W/32 \times 48$.

The decoder contains five decoder's blocks (DBs). Each of the first four DBs sequentially connects an up-sampling layer with a scaling factor of 2, a concatenation (Concate) operation, and two standard convolutional (Conv) layers with LReLUs. All the Conv layers in DBs are configured with dropout. The Concate operation in a DB stacks the feature cube from the upsampling layer and the one output by the LReLU in the corresponding EB. All Conv layers in the first four DBs have 96 output channels. The last DB contains three dropout Conv layers with LReLUs for mapping the feature cube back to an image of size $H \times W \times C$, and the numbers of output channels of these Conv layers are $64, 32, C$ respectively.

The architecture of our NN shares similarity with the ones used in some existing methods such as N2N [19]. The key differences are as follows. Firstly, we introduce dropout to the Conv layers in the decoder. In a dropout Conv layer, each weight entry is set to zero with a probability, and those untouched entries will be scaled for energy maintaining. Secondly, we use partial convolutions instead of the standard ones in the encoder, which further improves the effectiveness and efficiency of the NN training. See supplementary materials for more details of the partial convolution.

### 3.2. Training Scheme

As the NN is trained only on a single noisy image $\boldsymbol{y}$, we need to generate multiple image pairs from $\boldsymbol{y}$, which are different from $\boldsymbol{y}$ yet cover most of its information. With this goal, we generate a set of Bernoulli sampled instances of $\boldsymbol{y}$, denoted by $\{\widehat{\boldsymbol{y}}_m\}_{m=1}^M$. Recall that for $\boldsymbol{y}$, its Bernoulli sampled instance can be expressed as

$$\widehat{\boldsymbol{y}} := \boldsymbol{b} \odot \boldsymbol{y}, \tag{5}$$

where $\odot$ denotes the element-wise multiplication and $\boldsymbol{b}$ denotes one instance of binary Bernoulli vector whose entries are independently sampled from a Bernoulli distribution with probability $p \in (0, 1)$. Then, a set of image pairs $\{(\widehat{\boldsymbol{y}}_m, \overline{\boldsymbol{y}}_m)\}_{m=1}^M$ is defined as: for each $m$,

$$\widehat{\boldsymbol{y}}_m := \boldsymbol{b}_m \odot \boldsymbol{y}; \quad \overline{\boldsymbol{y}}_m := (\boldsymbol{1} - \boldsymbol{b}_m) \odot \boldsymbol{y}. \tag{6}$$

Given such a set of image pairs, the NN $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$, is trained by minimizing the following loss function:

$$\min_{\boldsymbol{\theta}} \sum_{m=1}^{M} \|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}_m) - \overline{\boldsymbol{y}}_m\|_{\boldsymbol{b}_m}^2, \tag{7}$$

where $\|\cdot\|_{\boldsymbol{b}}^2 = \|(\boldsymbol{1} - \boldsymbol{b}) \odot \cdot\|_2^2$. It can be seen that the loss of each pair is measured only on those pixels that are masked by $\boldsymbol{b}_m$. As the masked pixels are randomly selected using a Bernoulli process, the summation of the loss over all pairs measures the difference over all image pixels.

Clearly, the Bernoulli sampling we adopt can avoid the convergence of the NN to an identity mapping. Furthermore, training with the pairs of Bernoulli sampled instances $\{\widehat{\boldsymbol{y}}_m, \overline{\boldsymbol{y}}_m\}$ is very related to training with the pairs of a Bernoulli sampled instance $\widehat{\boldsymbol{y}}_m$ and the ground truth $\boldsymbol{x}$, especially when many such pairs are used for training. See the following proposition.

**Proposition 1.** Assume the noise components are independent and of zero mean. The expectation of the loss function (7) with respect to noise is the same as that of

$$\sum_{m=1}^{M} \|\mathcal{F}_{\boldsymbol{\theta}}(\widehat{\boldsymbol{y}}_m) - \boldsymbol{x}\|_{\boldsymbol{b}_m}^2 + \sum_{m=1}^{M} \|\boldsymbol{\sigma}\|_{\boldsymbol{b}_m}^2, \tag{8}$$

for arbitrary $\mathcal{F}_{\boldsymbol{\theta}}$, where $\boldsymbol{\sigma}(i)$ denotes the standard deviation of $\boldsymbol{n}(i)$.

*Proof.* See supplementary materials for the details. □

Since Bernoulli sampling can be viewed as an input layer with dropout, the use of Bernoulli sampled instances of noisy images can also be viewed as learning with dropout on single image. In the computation, we do not need to create the whole dataset of Bernoulli sampled instances in advance

but just enable dropout without energy scaling on the input layer and pass the copies of the input noisy images to the NN at each iteration. For further improvement, data augmentation is also used in the implementation by flipping the input image horizontally, vertically and diagonally. Thus, we have totally four versions of $\boldsymbol{y}$ for training.

### 3.3. Denoising Scheme

An NN trained with dropout provides a set of NNs whose certain weights follow independent Bernoulli distributions. The often-seen scheme for testing an NN with dropout is using the NN whose weights are scaled by their associated Bernoulli probability. As in our case, dropout is used for reducing the variance of the prediction, we propose to generate multiple NNs from the trained NN so as to have multiple estimators with likely certain degree of independence.

For denoising, multiple NNs $\mathcal{F}_{\boldsymbol{\theta}_1}, \cdots, \mathcal{F}_{\boldsymbol{\theta}_N}$ are formed by running dropout on the configured layers of the trained NN $\mathcal{F}_{\boldsymbol{\theta}^*}$. Then, multiple recovered images $\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_N$ are generated by feeding a Bernoulli sampled instance of $\boldsymbol{y}$ to each of the newly-formed NNs. The recovered images are then averaged to the obtain the final result $\boldsymbol{x}^*$:

$$\boldsymbol{x}^* = \frac{1}{N}\sum_{n=1}^{N} \hat{\boldsymbol{x}}_n = \frac{1}{N}\sum_{n=1}^{N} \mathcal{F}_{\boldsymbol{\theta}_n}(\boldsymbol{b}_{M+n} \odot \boldsymbol{y}). \quad (9)$$

In implementation, the above process can be done by simply collecting the results of stochastic forward passes through the trained model $\mathcal{F}_{\boldsymbol{\theta}^*}$. Furthermore, such forward passes can be done concurrently [12], resulting in constant running time identical to that of the standard dropout.

## 4. Experiments

The proposed method is evaluated on several denoising tasks: including blind Gaussian denoising, real-world noisy image denoising and salt-and-pepper noise removal. Due to space limitation, we only show partial results in this section. More results can be found in our supplementary materials.

### 4.1. Implementation Details

Throughout the experiments, all the PConv layers and Conv layers are with kernel sizes of $3 \times 3$, strides of 1, and zero padding of length 2. The hyper-parameter of each LReLU is set to 0.1. All the dropouts are conducted element-wisely with the dropout probability set to 0.3. The probability of Bernoulli sampling is also set to 0.3. The Adam optimizer is used for training. The learning rate is initialized to $10^{-5}$ with $4.5\times10^5$ training steps. During test, we use dropout 50 times to generate the final result. With parallel computation enabled on processing multiple images simultaneously, our implementation takes around 1.2 hours to process an image of size $256 \times 256$ on average using an RTX 2080Ti GPU. Our code will be released on GitHub.

### 4.2. Blind Gaussian Denoising

Two datasets are used for the performance evaluation in the case of additive white Gaussian noise (AWGN), including Set9 used in [25] with 9 color images and BSD68 used in [15] with 68 gray-scale images. Our experiments follow [25, 15] with more trials on high noise levels. Images are corrupted by the AWGN with noise levels: $\sigma = 25, 50, 75, 100$ for Set9 and $\sigma = 25, 50$ for BSD68.

**Comparison to single-image-based methods.** Several representative single-image-based denoising methods with published codes are selected for comparison: KSVD [11], PALM-DL [2], (C)BM3D [10] and DIP [25]. (C)BM3D is a well-known non-local method, KSVD and PALM-DL are two dictionary-learning-based methods, and DIP is an unsupervised deep-learning-based method. (C)BM3D, KSVD and PALM-DL are non-blind to the noise level, while DIP is blind if stopped with a universal maximal iteration number. However, we found that DIP's performance is sensitive to the iteration number for different noise levels and it becomes much better if the iteration is stopped once the residual matches the given noise level. Thus, we use such a non-blind version of DIP, denoted by DIP*, for comparison. Also, our method is compared to the single-image extension of N2V and N2S, denoted by N2V(1) and N2S(1), using their codes from the papers' GitHub sites. Note that N2V(1), N2S(1) and ours are blind to the noise level.

See Table 1 and Fig. 2 for the comparison. (a) Not surprisingly, our method outperforms KSVD and PALM-DL with a large margin, which is attributed to the advantage of deep learning over dictionary learning. (b) In comparison to the single-image-learning based denoising NNs, including DIP*, N2V(1) and N2S(1), ours also performs much better on all noise levels. This shows the effectiveness of using our dropout-based ensemble in test. (c) In comparison to one top performer in non-learning methods, (C)BM3D, our method performs better on all other noise levels.

**Comparison to dataset-based deep learning methods.** Our method is also compared to several recent dataset-based deep learning methods with published training codes, including N2V [15], N2S [3], N2N [19] and (C)DnCNN [31]. Recall that N2V and N2S are trained on unorganized noisy images, N2N is trained on paired noisy images, and (C)DnCNN is trained on clean/noisy image pairs. Following N2V's setting and its noisy data generation scheme, we train N2V and N2S on CBSD300 [15] and CBSD300's gray-scale version for color/gray-scale image denoising respectively. Regarding N2N, we use its published model trained on color images with the noise level range $\mathbb{L} = [0, 50]$ for the test on Set9 with $\sigma = 25, 50$. For other settings, we train N2N's model using CBSD300 with N2N's noisy image pair generation scheme. For (C)DnCNN, we use its pre-trained model on $\mathbb{L} = [0, 55]$ for the test with

Table 1. Average PSNR(dB)/SSIM(1.00E-1) of AWGN removal results on Set9 and BSD68. The best results in all approaches are marked in bold, and the best ones in single-image-based approaches or dataset-based deep approaches are underlined.

| Dataset | $\sigma$ | Single-image learning or non-learning methods | | | | | | | Dataset-based deep learning methods | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | KSVD | PALM-DL | (C)BM3D | N2V(1) | N2S(1) | DIP* | Ours | N2V | N2S | N2N | (C)DnCNN |
| Set9 | 25 | 30.00/9.35 | 29.84/9.32 | 31.67/9.55 | 28.12/9.12 | 29.30/9.40 | 30.77/9.42 | **31.74/9.56** | 30.66/9.47 | 30.05/9.44 | 31.33/9.57 | 31.42/9.56 |
| | 50 | 26.50/8.70 | 26.64/8.70 | 28.95/9.22 | 26.01/8.75 | 27.25/9.04 | 28.23/9.10 | **29.25/9.28** | 27.81/9.12 | 27.51/9.05 | 28.94/9.29 | 28.84/9.25 |
| | 75 | 24.29/8.10 | 24.55/8.12 | 27.36/8.95 | 24.18/8.27 | 25.85/8.61 | 26.64/8.83 | **27.61/9.01** | 25.99/8.75 | 26.49/8.82 | 27.42/9.05 | 27.36/9.01 |
| | 100 | 23.12/7.70 | 23.18/7.67 | 26.04/8.68 | 23.55/7.80 | 24.67/8.48 | 25.41/8.58 | 26.27/8.77 | 25.37/8.58 | 25.46/8.57 | **26.45/8.86** | 26.30/8.78 |
| BSD68 | 25 | 28.42/7.96 | 28.24/7.90 | 28.56/8.01 | 25.34/6.81 | 27.19/7.69 | 27.96/7.74 | 28.70/8.03 | 27.72/7.94 | 28.12/7.92 | 28.86/8.23 | **29.14/8.22** |
| | 50 | 25.08/6.53 | 25.09/6.49 | 25.62/6.87 | 23.85/6.18 | 24.53/6.42 | 25.04/6.45 | 25.92/6.99 | 25.12/6.84 | 25.62/6.78 | 25.77/7.00 | **26.20/7.15** |

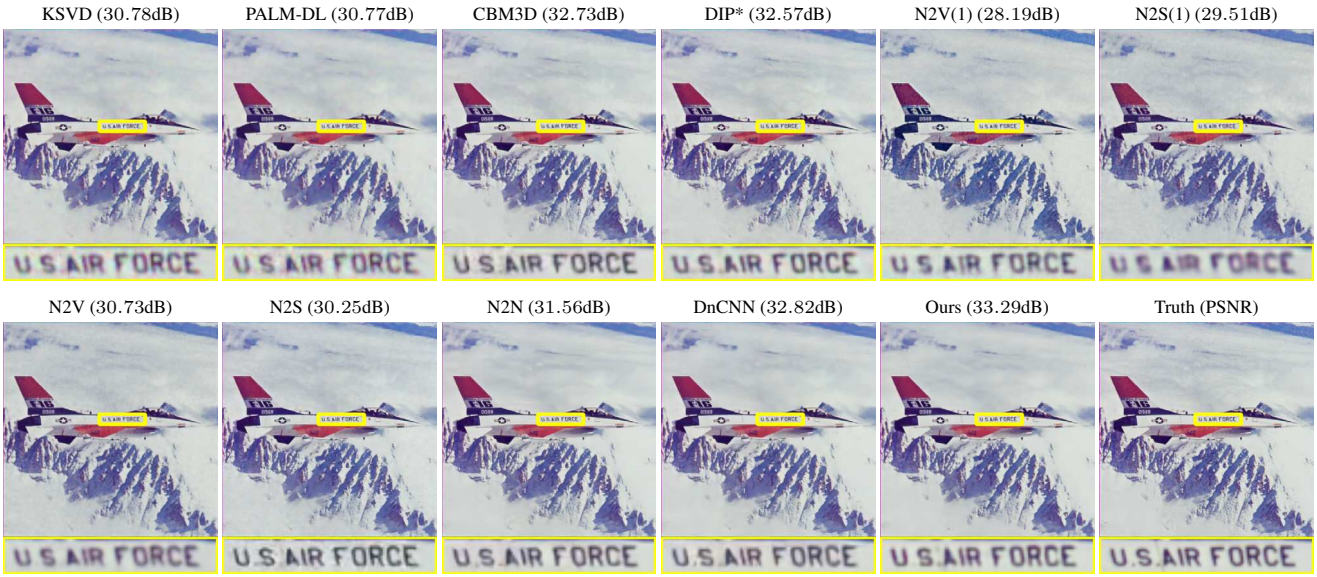


Figure 2. Visual results of blind AWGN denoising on image 'F16' of Set9 with noise level $\sigma = 25$.

$\sigma = 25, 50$ and retrain its model on $\mathbb{L} = [55, 110]$ using CBSD500 [21] for $\sigma = 75, 100$.

See Table 1 for the comparison. (a) As expected, deep learning benefits a lot from sufficient high-quality training data with noisy/clean image pairs, and (C)DnCNN is the top performer in BSD68. (b) It is surprising that our method performs much better than N2V and N2S which are trained with unorganized training samples. One reason might be that the unorganized training samples do not provide accurate information of the truth for the noisy image being processed. Oppositely, as the training data varies with different patterns and different noise levels, it might introduce misleading unrelated features into the NN. In contrast, our method avoids such an issue, as the training is on the noisy image being processed. (c) Very surprisingly, our method even outperforms N2N and DnCNN in many scenarios, despite the fact they are trained over the dataset with paired samples, and ours is trained with only a single noisy image.

### 4.3. Removing Real-World Image Noise

The performance evaluation on real-world noisy image denoising is conducted on the PolyU dataset [28] with 100 real clean/noisy color image pairs. Our method is compared with CBM3D, TWSC [29], DIP, N2V, N2S and CDnCNN. We randomly select 70 images for training N2V, N2S and DnCNN, and the remaining images are used for test. These NNs are trained using their published codes with our effort on parameter tuning-up. The noise level is estimated by the method [7] for CBM3D.

Table 2. Average PSNR(dB)/SSIM results on PolyU.

| Metric | CBM3D | TWSC | DIP | N2V | N2S | CDnCNN | Ours |
|---|---|---|---|---|---|---|---|
| PSNR | 36.98 | 36.10 | 36.95 | 34.08 | 35.46 | **37.55** | 37.52 |
| SSIM | 0.977 | 0.963 | 0.975 | 0.954 | 0.965 | **0.983** | 0.980 |

See Table 2 for the quantitative evaluation. Our method performs better than the non-learning methods including BM3D and TWSC, which shows the power of deep learning. Furthermore, except CDnCNN, our method noticeably outperforms other deep-learning-based methods, either the single-image-based or dataset-based ones. The reason for our superior results might be that the content of training samples is quite diverse, and thus the training samples and target images are not strongly correlated. Such a weak cor-


Figure 3. Denoising results on a real-world noisy image.

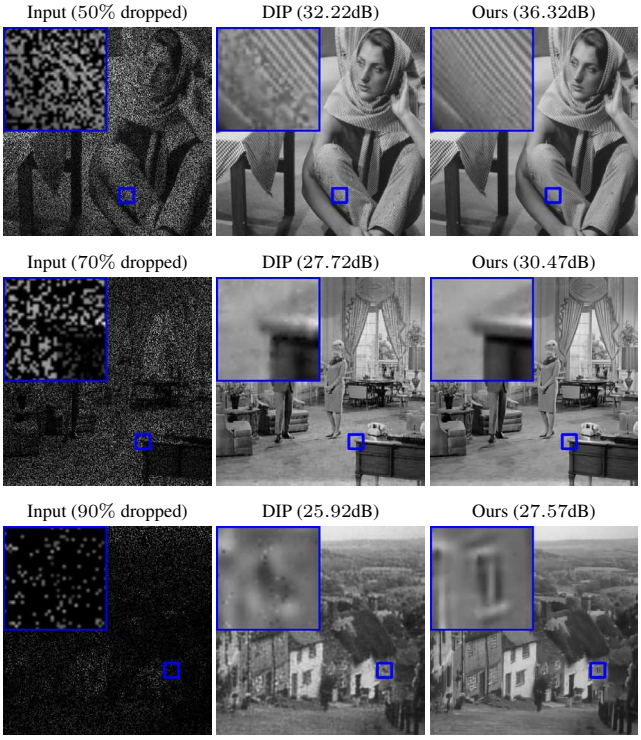


Figure 4. Visualization of removing (inpainting) pepper noise .

relation between training data and test images might mislead the NNs. Note that our quantitative results are very close to that of CDnCNN, and on some images our results are even better. See Fig. 3 for some visual comparison.

### 4.4. Removing Salt-and-Pepper Noise and Beyond

Removing salt-and-pepper noise (impulse noise) from images can be cast as inpainting randomly-missing image pixels. Following DIP, we use the Set11 dataset [25] for the performance evaluation on inpainting (*i.e.* non-blindly removing salt-and-pepper noise). As pixel values are completely erased by the salt-and-pepper noise, we only use uncorrupted pixels to train the NN. That is, only running sampling on un-corrupted pixels for generating the Bernoulli sampled instances, and the loss is not measured on corrupted image pixels. To generate the corrupted images for evaluation, we randomly drop the pixels of each image with ratios 50%, 70% and 90% respectively. Besides DIP, we use CSC [22], a dictionary-learning-based inpainting method, for comparison. See Table 3 for the quantitative comparison. Our method is much better than DIP and CSC. See also Fig. 4 for the visual comparison on three images.

**Image inpainting.** Our method is also tested on inpainting missing image regions. See Fig. 5 for two demos. It can be seen that the image quality of our results is better than that of DIP. For instance, DIP produced faint text imprints around the nose, which is not the case in our result.

Table 3. Average PSNR(dB)/SSIM of inpainting results on Set11.

| Dropping Ratio | CSC | DIP | Ours |
|---|---|---|---|
| 50% | 32.97/0.912 | 33.48/0.930 | **35.14/0.954** |
| 70% | 28.44/0.855 | 28.50/0.848 | **31.06/0.897** |
| 90% | 24.34/0.712 | 24.24/0.727 | **25.91/0.792** |


Figure 5. Visual results of text/scribble inpainting.

### 4.5. Ablation Study

To evaluate the effectiveness of its individual components, the following ablation studies on our method are conducted on the Set9 dataset with $\sigma = 25$. (a) w/o dropout:

disabling dropout on all layers during training and test; (b) w/o ensemble: using the trained dropout NN directly without dropout-based ensemble in test; (c) w/o sampling: using the original input image without Bernoulli sampling; (d) w/o PConv: replacing all PConv layers with Conv layers .

See Table 4 for the comparison, which leads to the following conclusions. (a) The comparison of 'Ours' v.s. 'w/o dropout', shows the important role of dropout in our method, as it causes significant PSNR drop, around 7.3dB, if no dropout is involved in either training or test. (b) Training with dropout itself is critical when training an NN with a single image, as the comparison of 'w/o ensemble' vs. 'w/o dropout', shows that only using dropout in training also leads to significant improvement. It justifies that dropout greatly helps overcoming the overfitting problem in our setting. (c) The comparison of 'Ours' vs. 'w/o ensemble', shows that running dropout in test is important, as it brings around 1.7dB gain in PSNR. It justifies the effectiveness of dropout-based ensemble in variance reduction. (d) The results of 'w/o sampling' show the importance of using Bernoulli sampling instances for training samples, which is consistent to what is observed in N2V and N2S. (e) The results of 'w/o PConv' show that partial convolution has a minor but worthwhile contribution to the performance.

Table 4. Results of ablation studies on Set9 with $\sigma = 25$.

| Ablation (**w/o**) | dropout | ensemble | sampling | PConv | Ours |
|---|---|---|---|---|---|
| PSNR(dB) | 23.88 | 29.92 | 23.12 | 31.26 | **31.74** |
| SSIM | 0.658 | 0.932 | 0.744 | 0.938 | **0.956** |

### 4.6. More Analysis

**Behavior of dropout-based ensemble.** In Fig. 6, we show how the prediction times, the value of $N$ in (9), impact the denoising performance on two sample images. We can see that the PSNR value increases as more predictions are used for averaging during test. The performance gain saturates when sufficient predictions are used. Thus, our trained NN with dropout in test can produce quite independent results such that their average is capable of effectively reducing the variance of prediction.

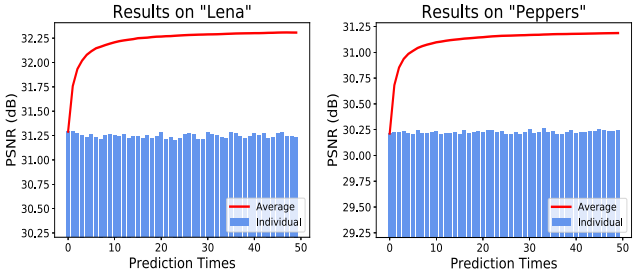


Figure 6. PSNR versus prediction times. Blue bars denote the PSNR results of individual inferences, and red curves denote the cumulative average PSNR.

**Stability over iterations.** As mentioned previously, DIP's performance is sensitive to the iteration number. It can be seen from Fig. 7 that DIP has its optimal performance happening at different steps (gray points) for different images, and its performance may have noticeable drop after passing the optimal step. In contrast, Fig. 7 shows that the performance of our method keeps unaffected after sufficient training steps. Such a feature is attractive for practical use as it requires little manual intervention.

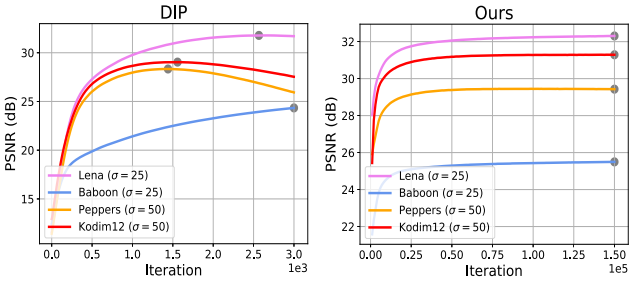


Figure 7. PSNR versus number of training iterations.

## 5. Conclusion

We proposed Self2Self, a self-supervised deep learning method for image denoising, which only uses the input noisy image itself for training and thus has no prerequisite on the training data collection. Our recipe for reducing the variance of prediction when training a denoising NN on a single noisy image is a dropout-based scheme. Dropouts are used during training as well as test, in terms of both dropping nodes in the NN and dropping pixels (Bernoulli sampling) in the input noisy image. This brings about different estimates of the ground truth image, which are averaged to yield the final output with reduced variance of prediction. Extensive experiments showed that, the performance of our denoising NN trained by the proposed Self2Self scheme is much better than that of other non-learning-based denoisers and single-image-learning denoisers. It is even close to that of those dataset-based deep learning methods. The results presented in this paper can inspire further investigations on self-supervised learning techniques in image recovery.

## Acknowledgment

Yuhui Quan would like to acknowledge the support from National Natural Science Foundation of China (61872151, U1611461), Natural Science Foundation of Guangdong Province (2017A030313376) and Fundamental Research Funds for Central Universities of China (x2js-D2181690). Tongyao Pang and Hui Ji would like to acknowledge the support from Singapore MOE Academic Research Fund (AcRF) Tier 2 research project (MOE2017-T2-2-156).

## References


[1] Chenglong Bao, Jian-Feng Cai, and Hui Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *Proc. ICCV*, pages 3384–3391, 2013. 3

[2] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *Trans. Pattern Anal. Mach. Intell.*, 38(7):1356–1369, 2015. 3, 5

[3] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. *Proc. ICML*, 2019. 1, 2, 3, 5

[4] Jian-Feng Cai, Hui Ji, Zuowei Shen, and Gui-Bo Ye. Data-driven tight frame construction and image denoising. *Appl.Comput. Harmonic Anal.*, 37(1):89–105, 2014. 3

[5] Sungmin Cha, Taeeon Park, and Taesup Moon. Gan2gan: Generative noise learning for blind image denoising with single noisy images. *arXiv preprint arXiv:1803.04189*, 2019. 3

[6] Antonin Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1-2):89–97, 2004. 3

[7] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In *Proc. ICCV*, pages 477–485, 2015. 6

[8] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proc. CVPR*, pages 3155–3164, 2018. 3

[9] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *Trans. Pattern Anal. Mach. Intell.*, 39(6):1256–1272, 2016. 3

[10] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007. 1, 3, 5

[11] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, 2006. 3, 5

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. ICML*, pages 1050–1059, 2016. 2, 5

[13] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proc. CVPR*, June 2019. 3

[14] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *Proc. CVPR*, pages 6054–6063, 2019. 3

[15] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proc. CVPR*, pages 2129–2137, 2019. 1, 2, 3, 5

[16] Alexander Krull, Tomas Vicar, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *arXiv preprint arXiv:1906.00651*, 2019. 3

[17] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Proc. NIPS*, pages 6968–6978, 2019. 3

[18] Stamatios Lefkimmiatis. Universal denoising networks: a novel cnn architecture for image denoising. In *Proc. CVPR*, pages 3204–3213, 2018. 3

[19] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *Proc. ICML*, 2018. 1, 3, 4, 5

[20] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proc. ECCV*, pages 85–100, 2018. 4

[21] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Iccv Vancouver:, 2001. 6

[22] Vardan Papyan, Yaniv Romano, Jeremias Sulam, and Michael Elad. Convolutional dictionary learning via local processing. In *Proc. ICCV*, pages 5296–5304, 2017. 3, 7

[23] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *Proc. CVPR*, pages 2774–2781, 2014. 3

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learning Research*, 15(1):1929–1958, 2014. 2

[25] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proc. CVPR*, pages 9446–9454, 2018. 1, 3, 5, 7

[26] Raviteja Vemulapalli, Oncel Tuzel, and Ming-Yu Liu. Deep gaussian conditional random field network: A model-based deep network for discriminative denoising. In *Proc. CVPR*, pages 4801–4809, 2016. 1, 3

[27] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Proc. NIPS*, pages 341–349, 2012. 3

[28] 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. 6

[29] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *Proc. ECCV*, pages 20–36, 2018. 6

[30] Xuhui Yang, Yong Xu, Yuhui Quan, and Hui Ji. Image denoising via sequential ensemble learning. *IEEE Trans. Image Process.*, 29(12):5038–5049, 2020. 3

[31] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017. 1, 3, 5

[32] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.*, 27(9):4608–4622, 2018. 1, 3