Training deep learning based denoisers without ground truth data

Shakarim Soltanayev    Se Young Chun
Department of Electrical Engineering
Ulsan National Institute of Science and Technology (UNIST)
Ulsan, Republic of Korea
{shakarim,sychun}@unist.ac.kr

Abstract

Recent deep learning based denoisers often outperform state-of-the-art conventional denoisers such as BM3D. They are typically trained to minimize the mean squared error (MSE) between the output of a deep neural network and the ground truth image. In deep learning based denoisers, it is important to use high quality noiseless ground truth for high performance, but it is often challenging or even infeasible to obtain such a clean image in application areas such as hyperspectral remote sensing and medical imaging. We propose a Stein’s Unbiased Risk Estimator (SURE) based method for training deep neural network denoisers only with noisy images. We demonstrated that our SURE based method without ground truth was able to train deep neural network denoisers to yield performance close to deep learning denoisers trained with ground truth and to outperform state-of-the-art BM3D. Further improvements were achieved by including noisy test images for training denoiser networks using our proposed SURE based method.

1 Introduction

Deep learning has been successful in various high-level computer vision tasks \[1\] such as image classification \[2,3\], object detection \[4,5\], and semantic segmentation \[6,7\]. Deep learning has also been investigated for low-level computer vision tasks such as image denoising \[8,9,10,11,12\], image inpainting \[13\], and image restoration \[14,15,16\]. In particular, image denoising is a fundamental computer vision task that does not only yield a clean image with reduced noise, but also improves other tasks such as image classification \[8\] and image restoration \[16\].

Deep learning based image denoisers \[9,11,12\] have yielded performance equivalent to or better than conventional state-of-the-art denoising techniques such as BM3D \[17\]. These deep learning based denoisers typically train their networks by minimizing the mean squared error (MSE) between the output of a network and the ground truth (noiseless) image. Thus, it is crucial to have high quality noiseless images for high performance deep learning denoisers. So far deep learning based image denoisers have been successful since high quality camera sensors and abundant light allows to obtain massive amount of high quality, almost noiseless 2D images in daily environment. Acquiring such a high quality photo is quite cheap these days with smart phones and digital cameras.

However, it is challenging to apply current deep learning based image denoisers with MSE to some application areas such as hyperspectral remote sensing and medical imaging where acquiring noiseless ground truth data is expensive or sometimes even infeasible. For example, hyperspectral imaging contains hundreds of spectral information per pixel so that noise in hyperspectral imaging sensors \[18\]. A long acquisition may improve image quality, but it is challenging to perform it with spaceborne or airborne hyperspectral imaging. Similarly, in medical imaging, high resolution 3D MRI (sub-mm resolution) often requires several hours of acquisition time for a single high quality volume, but
reducing acquisition time leads to increased noise. In X-ray CT, image noise can be substantially reduced by increasing radiation dose. Recent works on deep learning based image denoisers \cite{19, 20} utilized normal dose CT images as ground truth so that denoising networks were able to be trained to yield excellent performance. However, increased radiation dose leads to harmful effects in scanned subjects and too high dose may saturate CT detectors (e.g., similar to taking a photo of sun without any filter). Thus, acquiring ground truth data for newly developed CT scanners seems challenging without compromising subjects’ safety.

Conventional denoising methods do not usually require noiseless ground truth images to perform denoising, but often require them for tuning parameters of image filters for the best possible results (minimum MSE). In order to find the optimal parameters of conventional denoisers without ground truth data, there have been several works using Stein’s Unbiased Risk Estimator (SURE) \cite{21}, an unbiased estimator of MSE. For popular NLM (Non-Local Means) filter \cite{22}, the analytical form of SURE for NLM was used to optimize denoiser performance \cite{23, 24}. For denoisers whose analytical forms of SURE are not available, Ramani et al. \cite{25} proposed a Monte-Carlo based SURE (MC-SURE) method to determine near-optimal denoising parameters by brute-force search on parameter space. Deledalle et al. \cite{26} investigated on the approximation of a weak gradient of SURE to optimize parameters using quasi-Newton algorithm. However, since this method requires computing full weak Jacobian, it is not applicable for high dimensional parameter spaces such as deep neural networks.

We propose a SURE based training method for deep neural network denoisers without ground truth data. In Section 2, we review key results of SURE and MC-SURE. Then, in Section 3, we describe our proposed training method using MC-SURE and stochastic gradient for deep learning based image denoisers. In Section 4, simulation results are presented for conventional state-of-the-art denoiser (BM3D), deep learning based denoiser trained with BM3D as the ground truth, the same deep neural network denoiser with the proposed SURE training without the ground truth, and the same denoiser network with ground truth data as a reference. Lastly, in Section 5, we conclude this paper by discussing several potential issues for further studies.

2 Background

2.1 Stein’s unbiased risk estimator (SURE)

The signal (or image) with Gaussian noise can be modeled:

\[ y = x + n \]

where \( x \in \mathbb{R}^K \) is an unknown signal following \( x \sim p(x) \), \( y \in \mathbb{R}^K \) is a known measurement and \( n \in \mathbb{R}^K \) is an i.i.d. Gaussian noise such that \( n \sim N(0, \sigma^2 I) \) and \( I \) is an identity matrix. We denote \( n \sim N(0, \sigma^2 I) \) as \( n \sim N_0, \sigma^2 \). An estimator of \( x \) from \( y \) (or denoiser) can be defined as a function of \( y \) such that

\[ h(y) = y + g(y) \]

where \( h, g \) are functions from \( \mathbb{R}^K \) to \( \mathbb{R}^K \). Then, the SURE for \( h(y) \) can be derived as follows:

\[ \eta(h(y)) = \sigma^2 + \frac{\| g(y) \|^2}{K} + \frac{2\sigma^2}{K} \sum_{i=1}^{K} \frac{\partial g_i(y)}{\partial y_i} = \frac{\| y - h(y) \|^2}{K} - \sigma^2 + \frac{2\sigma^2}{K} \sum_{i=1}^{K} \frac{\partial h_i(y)}{\partial y_i} \]

where \( \eta : \mathbb{R}^K \to \mathbb{R} \) and \( y_i \) is the \( i \)th element of \( y \). Then, for a fixed \( x \), the following theorem holds:

**Theorem 1.** \[21, 27\] The random variable \( \eta(h(y)) \) is an unbiased estimator of

\[ \text{MSE}(h(y)) = \frac{1}{K} \| x - h(y) \|^2 \]

or

\[ \mathbb{E}_{n \sim N_0, \sigma^2} \left\{ \frac{\| x - h(y) \|^2}{K} \right\} = \mathbb{E}_{n \sim N_0, \sigma^2} \{ \eta(h(y)) \} \]

(4)
where $\mathbb{E}_{n \sim \mathcal{N}_0, x^2} \{ \cdot \}$ is the expectation in terms of the random vector $n$. Note that in Theorem 1 $x$ is treated as a fixed, deterministic vector.

In practice, $\sigma^2$ can be estimated [25] and $\| y - h(y) \|^2$ only requires the output of the estimator (or denoiser). The last divergence term of (3) can be obtained analytically for some special cases such as linear filters or NLM filters [24]. However, it is challenging to calculate this term for more general denoising methods analytically.

2.2 Monte-Carlo Stein’s unbiased risk estimator (MC-SURE)

Ramani et al. [25] introduced a fast Monte-Carlo approximation of the divergence term in (3) for general denoisers. For a fixed unknown true image $x$, the following theorem holds:

**Theorem 2.** [25] Let $\tilde{n} \sim \mathcal{N}_{0,1} \in \mathbb{R}^K$ be independent of $n, y$. Then,

$$
\sum_{i=1}^{K} \frac{\partial h_i(y)}{\partial y_i} = \lim_{\epsilon \to 0} \mathbb{E}_{\tilde{n}} \left\{ \tilde{n}^T \left( \frac{h(y + \epsilon \tilde{n}) - h(y)}{\epsilon} \right) \right\}
$$

(5)

provided that $h(y)$ admits a well-defined second-order Taylor expansion. If not, this is still valid in the weak sense provided that $h(y)$ is tempered.

Based on Theorem 2, the divergence term in (3) can be approximated with one realization of $\tilde{n} \sim \mathcal{N}_{0,1}$ and a fixed small positive value $\epsilon$:

$$
\frac{1}{K} \sum_{i=1}^{K} \frac{\partial h_i(y)}{\partial y_i} \approx \frac{1}{\epsilon K} \tilde{n}^T (h(y + \epsilon \tilde{n}) - h(y))
$$

(6)

where $t$ is a transpose operator. This expression has been shown to yield accurate unbiased estimate of MSE for many conventional denoising methods $h(y)$ [25].

3 Method

In this section, we will develop our proposed MC-SURE based training method for deep learning based denoisers without noiseless ground truth images assuming Gaussian noise model in (1).

3.1 Training denoisers using stochastic gradient method

A typical risk for image denoisers with the signal generation model (1) is

$$
\mathbb{E}_{x \sim p(x), n \sim \mathcal{N}_0, x^2} \| x - h(y; \theta) \|^2
$$

(7)

where $h(y; \theta)$ is a deep learning based denoiser parametrized with a large-scale vector $\theta$. It is usually infeasible to calculate (7) exactly due to expectation operator. Thus, the empirical risk for (7) is used as a cost function as follows:

$$
\frac{1}{N} \sum_{j=1}^{N} \| h(y^{(j)}; \theta) - x^{(j)} \|^2
$$

(8)

where $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(N)}, y^{(N)})\}$ are $N$ number of training data, sampled from the joint distribution of $x^{(j)} \sim p(x)$ and $n^{(j)} \sim \mathcal{N}_0, x^2$. Note that (8) is an unbiased estimator of (7).

To train the deep learning network $h(y; \theta)$ with respect to $\theta$, a gradient based optimization algorithm is used such as stochastic gradient descent (SGD) [28], momentum, Nesterov momentum [29], or Adam optimizer [30]. For any gradient based optimization method, it is essential to calculate the gradient of (7) with respect to $\theta$ as follows:

$$
\mathbb{E}_{x \sim p(x), n \sim \mathcal{N}_0, x^2} 2 \nabla \theta h(y; \theta)^T (h(y; \theta) - x).
$$

(9)

Therefore, it is sufficient to calculate the gradient of the empirical risk (8) to approximate (9) for any gradient based optimization.
In practice, calculating the gradient of (8) for large \( N \) is inefficient since a small amount of well-shuffled training data can often approximate the gradient of (8) well. Thus, a mini-batch is typically used for efficient deep neural network training by calculating the mini-batch empirical risk as follows:

\[
\frac{1}{M} \sum_{j=1}^{M} \| h(y^{(j)}; \theta) - x^{(j)} \|^2
\]  

(10)

The equation (10) is still an unbiased estimator of (7) provided that the training data is randomly permuted every epoch and the same data is used no more than once per each epoch.

### 3.2 Proposed training method for deep learning based denoisers

To incorporate MC-SURE into a stochastic gradient based optimization algorithm such as SGD or Adam optimizer for training, we modify the risk (7) as

\[
\mathbb{E}_{x \sim p(x)} \left[ \mathbb{E}_{n \sim \mathcal{N}_0,\sigma^2} \left( \| x - h(y; \theta) \|^2 \right) \right].
\]  

(11)

where (11) is equivalent to (7) due to conditioning.

From Theorem 1, an unbiased estimator for \( \mathbb{E}_{n \sim \mathcal{N}_0,\sigma^2} \left( \| x - h(y; \theta) \|^2 \right) \) can be derived as

\[
K \rho(h(y; \theta))
\]  

(12)

such that for a fixed \( x \),

\[
\mathbb{E}_{n \sim \mathcal{N}_0,\sigma^2} \left( \| x - h(y; \theta) \|^2 \right) = \mathbb{E}_{n \sim \mathcal{N}_0,\sigma^2} \| x - h(y; \theta) \|^2 = K \mathbb{E}_{n \sim \mathcal{N}_0,\sigma^2} \rho(h(y; \theta)).
\]

Then, using the empirical risk expression in (10), an unbiased estimator for (7) is

\[
\frac{1}{M} \sum_{j=1}^{M} \| y^{(j)} - h(y^{(j)}; \theta) \|^2 - K \sigma^2 + 2 \sigma^2 \sum_{i=1}^{K} \partial h_i(y^{(j)}; \theta)
\]  

(13)

where there is no noiseless ground truth data \( x^{(j)} \) used in (13).

Finally, the last divergence term in (13) can be approximated using MC-SURE so that the final unbiased risk estimator for (7) will be

\[
\frac{1}{M} \sum_{j=1}^{M} \| y^{(j)} - h(y^{(j)}; \theta) \|^2 - K \sigma^2 + \frac{2 \sigma^2}{\epsilon} (\tilde{n}^{(j)})^t \left( h(y^{(j)}; \theta) + \epsilon \tilde{n}^{(j)}; \theta \right) - h(y^{(j)}; \theta)
\]  

(14)

where \( \epsilon \) is a small fixed positive number and \( \tilde{n}^{(j)} \) is a single realization from the standard normal distribution for each training data \( j \). In order to make sure that the estimator (14) is unbiased, the order of \( y^{(j)} \) should be randomly permuted and the new set of \( \tilde{n}^{(j)} \) should be generated every epoch.

The implementation of deep learning based image denoiser with the cost function of (14) can be done using deep learning development framework such as TensorFlow by defining the cost function properly. Then, the gradient of (14) can be automatically calculated when the training is performed.

One potential advantage of our SURE based training method is that we can use all available data without noiseless ground truth images. In other words, we can train denoising deep neural networks not only with training data, but also with test data together. This advantage may further improve the performance of deep learning based denoisers.

Lastly, almost any deep learning based image denoiser can utilize our MC-SURE based training by modifying the cost function from (10) to (14) as far as it satisfies the condition in Theorem 2. Many deep learning based denoisers with differentiable activation functions (e.g., sigmoid) can meet this condition. Some denoisers with piecewise differentiable activation functions (e.g., ReLU) still can utilize Theorem 2 in the weak sense since

\[
\| h(y; \theta) \| \leq C_0 (1 + \| y \|^{n_0})
\]

for some \( n_0 > 1 \) and \( C_0 > 0 \). Therefore, we expect that our proposed method should work for most deep learning image denoisers [8, 9, 10, 11, 12].
4 Simulation results

In this section, denoising simulation results with MNIST dataset using a simple stacked denoising autoencoder (SDA) [8] and large-scale natural image dataset using deep convolutional neural network (CNN) image denoiser (DnCNN) [11] will be presented. For both cases, conventional state-of-the-art image denoiser, BM3D [17], was also tested.

All of the networks presented in this section (called NET, which can be one among SDA and DnCNN) were trained using one of the following two optimization objectives: (MSE) the minimum MSE between denoised image and its ground truth image, \( (10) \); (SURE) our proposed minimum MC-SURE without ground truth, \( (14) \). NET-MSE methods generated noisy training images every epochs following \( (11) \), while our proposed NET-SURE methods used noisy images only that were generated once before training. We also propose SURE-T method which utilizes noisy test images along with noisy training images and without ground truth data. Table 1 summarizes all simulated configurations including conventional state-of-the-art image denoiser, BM3D [17], that does not require any training.

| Method       | Description                                                                        |
|-------------|------------------------------------------------------------------------------------|
| BM3D        | Conventional method                                                                |
| NET-BM3D    | Optimizing MSE with BM3D output as ground truth                                     |
| NET-SURE    | Optimizing SURE without ground truth                                               |
| NET-SURE-T  | Optimizing SURE without ground truth, but with noisy test data                     |
| NET-MSE-GT  | Optimizing MSE with ground truth                                                   |

4.1 Results: MNIST dataset

We performed denoising simulations with MNIST dataset. The noisy images were generated based on the model \( (1) \) with two noise levels (one \( \sigma = 25 \) and the other with \( \sigma = 50 \)). For the experiments on the MNIST dataset with \( 28 \times 28 \) pixels, a simple SDA was chosen [8]. Decoder and encoder networks each consists of two convolutional layers (kernel size \( 3 \times 3 \)) with sigmoid activation functions each having a stride of 2 (both conv and conv transposed). Thus, a training sample of size \( 28 \times 28 \) is downsampled to \( 7 \times 7 \) and then upsampled back to \( 28 \times 28 \).

SDA was trained to output a denoised image using a set of 55,000 training and 5,000 validation images. The performance of the model was tested with 100 images chosen randomly from the default test set of 10,000 images. For all cases, SDA was trained with Adam optimizer [30] with the learning rate of 0.001 for 100 epochs. Batch size was set to 200 and bigger batch sizes than that did not improve the performance. The \( \epsilon \) value in \( (6) \) was set to 0.0001.

Our proposed SDA-SURE yielded comparable performance to SDA-MSE-GT (only 0.01-0.03 dB difference) and outperformed conventional BM3D for all simulated noise levels, \( \sigma = 25, 50 \), as shown in Table 2. Further improvements were obtained by utilizing noisy test images in training (SDA-SURE-T) by 0.13 dB and 0.16 dB better than SDA-MSE-GT and SDA-SURE, respectively, at \( \sigma = 25 \). However, including noisy test data yielded almost the same performance compared to SDA-MSE-GT and SDA-SURE.

Figure 1 illustrates visual quality of simulated denoising methods at high noise level (\( \sigma = 50 \)). All SDA based methods clearly outperform conventional BM3D visually (BM3D image looks blurry compared to other SDA based results), while it is indistinguishable for the simulation results among all SDA methods with different cost function and training sets. These observations were confirmed by

| Methods | BM3D | SDA-REG | SDA-SURE | SDA-SURE-T | SDA-MSE-GT |
|---------|------|---------|----------|------------|------------|
| \( \sigma = 25 \) | 27.53 | 25.07   | 27.90    | 28.06      | 27.93      |
| \( \sigma = 50 \) | 21.82 | 19.85   | 25.23    | 25.24      | 25.24      |
the quantitative results shown in Table 2. All SDA based methods outperformed BM3D significantly, but there were very small differences among all SDA methods, even when using noisy test data.

4.2 Regularization effect of SDA

Parametrization of deep neural networks with different number of parameters and structures may introduce regularization effect in training denoisers. We further investigated this by training SDA to minimize the MSE between the output of SDA and input noisy image to explore its regularization effect (SDA-REG). In case of the noise level of \( \sigma = 50 \), early stopping was applied when the network started to overfit the noisy dataset after the first few epochs. The performance of this method was significantly worse than all the other methods with PSNR of 25.07 dB (\( \sigma = 25 \)) and 19.85 dB (\( \sigma = 50 \)) as shown in Table 2, that is about 2 dB lower than PSNR of BM3D. Noise patterns are visible as shown in Figure 1. This shows that the good performance of SDA does not arise just from its structure, but comes from optimizing MSE and SURE.

4.3 Accuracy of MC-SURE approximation

A small value must be assigned to \( \epsilon \) in (6) for accurate estimation of SURE. Ramani et al. [25] have observed that \( \epsilon \) can take a wide range of values and its choice is not critical. The admissible range of \( \epsilon \), however, depends on \( h_i(y; \theta) \). According to our preliminary experiments for SDA with MNIST dataset, any choice for \( \epsilon \) in between \([10^{-2}, 10^{-7}]\) worked well so that the SURE approximation matches close to the MSE during training as illustrated in Figure 2 (middle). Extremely small values \( \epsilon < 10^{-8} \) result in numerical instabilities as can be seen in Figure 2 (right). On the contrary, when \( \epsilon > 10^{-1} \), the approximation in (6) becomes inaccurate. Note, that these values are only for SDA trained with MNIST dataset. A suitable \( \epsilon \) value must be carefully selected for other cases such as DnCNN with large-scale parameters and high resolution images for high performance.

4.4 Results: high resolution natural images

To demonstrate the capabilities of SURE based deep learning denoisers, we investigated a deeper and more powerful denoising network called DnCNN [11] for high resolution images. DnCNN consists of 17-layers of CNN with batch normalization and ReLU activation functions. Each convolutional layer has 64 filters of size \( 3 \times 3 \). Following [11], the network was trained with 400 images of size \( 180 \times 180 \). In total 1772 \( \times \) 128 image patches of size \( 40 \times 40 \) were extracted randomly from those images. As in [11], two test sets were used to evaluate the performance: 12 widely used images (Set12) [17] and BSD68 dataset. For DnCNN-SURE-T, additional \( 808 \times 128 \) image patches were extracted from those noisy test images and were added to the training dataset. For all cases, the network was trained for 50 epochs using Adam optimizer with initial learning rate of 0.001 and then decayed to 0.0001 after 40 epochs. Batch size was set to 128 and bigger batch sizes than that did not improve the performance. Images were corrupted by three noise levels \( \sigma = 25, 50, 75 \).

DnCNN used residual learning [11] where the network is forced to learn the difference between noisy and ground truth images. Then, the output residual image was subtracted from the input noisy image to yield the estimated image. Therefore, for the case of residual learning, our network was trained with SURE using (15) as follows:

\[
h(y; \theta) = C \text{NN}_\theta(y) - y
\]

where \( C \text{NN}_\theta(.) \) is the DnCNN that is being trained using residual learning.

![Figure 1: Denoising results of SDA with various methods for MNIST dataset with \( \sigma=50 \) noise level](image-url)
Figure 2: Loss curves for the training SDA with MSE (blue) and its corresponding Monte-Carlo SURE (red) using different epsilon values, $\epsilon = 1$ (left), $\epsilon = 10^{-5}$ (middle) and $\epsilon = 10^{-8}$ (right). Monte-Carlo SURE well approximates true MSE for a wide range of $\epsilon$.

Table 3: Results of denoising methods on 12 widely used images (Set12) (performance in dB).

| IMAGE | C. MAN | HOUSE | PEPPERS | STARFISH | MONARCH | AIRPLANE | PARROT | LENA | BARBARA | BOAT | MAN | COUPLE | Average |
|-------|--------|-------|---------|----------|---------|----------|--------|------|---------|------|-----|--------|---------|
| BM3D  | 29.47  | 33.00 | 30.23   | 28.58    | 29.35   | 28.37    | 28.89  | 32.06| 30.64   | 29.78| 29.60| 29.97  | 30.64   |
| DnCNN-BM3D | 29.34 | 31.99 | 30.13   | 28.38    | 29.21   | 28.46    | 28.94  | 31.53| 28.89   | 29.8  | 29.5  | 29.8   | 30.06   |
| DnCNN-SURE | 29.80 | 32.70 | 30.58   | 29.08    | 30.11   | 29.24    | 29.17  | 32.06| 29.16   | 29.84| 29.89| 30.76  | 30.09   |
| DnCNN-SURE-T | 29.86 | 32.73 | 30.57   | 29.11    | 30.13   | 29.26    | 32.02  | 30.84| 29.91   | 29.86| 29.91| 29.78  | 30.14   |
| DnCNN-MSE-GT | 30.14 | 33.16 | 30.84   | 29.24    | 30.45   | 29.11    | 29.38  | 32.44| 29.91   | 31.11| 30.08| 30.68  | 30.42   |
| BM3D  | 26.00  | 29.51 | 26.58   | 26.01    | 25.78   | 25.15    | 25.98  | 28.93| 27.19   | 26.62| 26.79| 26.46  | 26.67   |
| DnCNN-BM3D | 25.76 | 28.43 | 26.3    | 24.9     | 25.66   | 25.15    | 25.82  | 28.36| 25.3    | 26.5  | 26.6  | 26.17  | 26.26   |
| DnCNN-SURE | 26.48 | 29.14 | 26.77   | 25.38    | 26.50   | 25.66    | 26.21  | 28.79| 24.86   | 26.78| 26.97| 25.51  | 26.87   |
| DnCNN-SURE-T | 26.47 | 29.20 | 26.78   | 25.39    | 26.53   | 25.65    | 26.21  | 28.81| 25.23   | 26.79| 26.97| 26.48  | 26.71   |
| DnCNN-MSE-GT | 27.03 | 29.92 | 27.21   | 25.65    | 28.95   | 25.93    | 26.43  | 29.31| 26.17   | 27.12| 27.22| 26.94  | 27.16   |
| BM3D  | 24.58  | 27.45 | 24.69   | 23.19    | 23.81   | 23.38    | 24.22  | 27.14| 25.08   | 25.03| 25.3  | 24.73  | 24.89   |
| DnCNN-BM3D | 24.11 | 27.02 | 24.48   | 23.09    | 23.73   | 23.40    | 24.06  | 27.11| 23.80   | 24.84| 25.19| 24.59  | 24.62   |
| DnCNN-SURE | 24.65 | 27.16 | 24.40   | 23.25    | 24.10   | 23.52    | 24.13  | 26.92| 23.02   | 25.09| 25.37| 24.70  | 24.70   |
| DnCNN-SURE-T | 24.82 | 27.34 | 24.56   | 23.34    | 24.25   | 23.56    | 24.48  | 27.03| 23.07   | 25.17| 25.45| 24.78  | 24.82   |
| DnCNN-MSE-GT | 25.46 | 28.04 | 25.22   | 23.62    | 24.81   | 23.97    | 24.72  | 27.80| 23.88   | 25.53| 25.68| 25.13  | 25.30   |

On a NVidia Titan X GPU, the training process took approximately 7 hours for DnCNN-MSE-GT and about 11 hours for DnCNN-SURE. SURE based method take more training time than MSE based method because of the additional divergence calculations done to optimize MC-SURE cost function. For DnCNN-SURE-T method, it took around 15 hours to complete the training due to larger dataset.

For DnCNN, selection of the $\epsilon$ value in (6) turned out to be important for denoising performance. To achieve a stable training with good performance $\epsilon$ had to be tuned for each of the noise levels $\sigma = 25$, 50, 75. We observed that the optimal value of $\epsilon$ was proportional to $\sigma$ as in [26]. All the experiments were performed setting $\epsilon = \sigma \times 10^{-4}$.

Tables 3 and 4 present denoising performance using BM3D [17], state-of-the-art deep CNN (DnCNN) image denoiser trained with MSE [11] and the same DnCNN image denoiser trained with SURE without knowing noiseless ground truth images for different dataset variations (as in Table 1). MSE based DnCNN image denoiser with ground truth data, DnCNN-MSE-GT, yielded the best denoising performance over other methods such as BM3D, which is consistent with the results in [11].

As we can see from Table 3, for Set12 dataset SURE based denoisers achieve comparable or better performance than BM3D for noise levels $\sigma = 25, 50$, while for higher noise level $\sigma = 75$, DnCNN-SURE and DnCNN-SURE-T had 0.19 dB and 0.07 dB lower average PSNR than BM3D. DnCNN-SURE-T outperformed DnCNN-SURE in all cases and had considerably better performance on some images such as ‘Barbara’. BM3D had exceptionally good denoising performance on the ‘Barbara’ image (up to 2.33 dB better PSNR) even outperforming DnCNN-MSE-GT method.

Table 4: Results of denoising methods on BSD68 dataset (performance in dB).

| Methods | BM3D | DnCNN-BM3D | DnCNN-SURE | DnCNN-SURE-T | DnCNN-MSE-GT |
|---------|------|------------|------------|--------------|--------------|
| $\sigma = 25$ | 28.56 | 28.54 | 28.97 | 29.00 | 29.20 |
| $\sigma = 50$ | 25.62 | 25.44 | 25.93 | 25.95 | 26.22 |
| $\sigma = 75$ | 24.20 | 24.09 | 24.31 | 24.37 | 24.66 |
In case of the BSD68 dataset (Table 4) SURE based methods outperformed BM3D for all the noise levels. Unlike in the case of Set12 images, here we observe that DnCNN-SURE has significantly better performance than BM3D ranging from 0.11 dB to 0.41 dB higher average PSNR. It can be observed that DnCNN-SURE-T benefits from the utilization of noisy test images slightly improving the average PSNR of DnCNN-SURE.

Difference between the performance of denoisers in Tables 3 and 4 can be explained by the working principle of BM3D. Since BM3D looks for similar image patches for denoising, repeated patterns (as in ‘Barbara’ image) and flat areas (as in ‘House’ image) can be a key factor to yield strong denoising results. One advantage of DnCNN-SURE over BM3D is that it doesn’t suffer from rare patch effect. If test image is relatively detailed and does not contain many similar patterns, BM3D will have poorer performance than our DnCNN-SURE method.

Figure 3 illustrates denoising results for an image from BSD68 dataset. BM3D yielded blurrier image than deep learning based denoisers and thus had worse PSNR. DnCNN-MSE-GT had the best denoised image with PSNR of 26.85, while both SURE methods had similar performance.

Two more methods were experimented on DnCNN. First was a hybrid method that involved pretraining with DnCNN-MSE-GT and fine-tuning with SURE using the noisy test images. However, this hybrid method could not outperform SURE based methods converging on a slightly lower average PSNR values. In the second method called DnCNN-BM3D, denoised images by BM3D were used as ground truth data and DnCNN was trained by optimizing MSE. This method also did not require noiseless images, however had the worst performance among all denoising methods including BM3D as shown in Tables 3 and 4.

5 Discussion & Conclusion

We proposed a MC-SURE based training method for general deep learning denoisers. Our proposed method train denoisers without noiseless ground truth data such that they have comparable denoising performance to conventional state-of-the-art BM3D or to the same denoisers that are trained with noiseless ground truth data. Our SURE based training method does not only work for simple SDA [8], but also works for state-of-the-art DnCNN network [11] without ground truth images.

In this work, Gaussian noise with known variance was assumed in all simulations. However, there are several noise estimation methods that can be used with SURE (see [25] for details). SURE can incorporate a variety of noise distributions other than Gaussian noise. For example, SURE for Poisson noise has been used for parameter selection of conventional filter [26]. Generalized SURE for exponential families has been proposed [32] so that other common noise in imaging systems can be potentially considered for SURE based methods.

Note that SURE does not require any prior knowledge on images, so it can potentially be applied to the measurement domain for different application areas such as medical imaging. If measurement and image domains are different (e.g., Fourier transform, Radon transform), then the noise property in image domain should be carefully investigated in order to use our proposed method in image domain. Since SURE expression is the sum of all unbiased risk estimator for each pixel, it is trivial to apply SURE locally with spatially varying variance. However, due to correlation in noise of image domain
(through radon transform for CT or PET, or through Fourier transform for MRI), further investigation will be necessary to apply our proposed method to image domain directly.

Our proposed SURE based deep learning denoiser can potentially be useful for applications with massive amount of noisy images, but with few noiseless images or with expensive noiseless images. Deep learning based denoising research is still evolving, it may be even possible to achieve significantly better performance than BM3D or other conventional state-of-the-art denoisers with our SURE based training method. Further investigation will be needed for high performance denoising networks.

Acknowledgments

This work was partly supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B05035810) and the Technology Innovation Program or Industrial Strategic Technology Development Program (10077533, Development of robotic manipulation algorithm for grasping/assembling with the machine learning using visual and tactile sensing information) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, May 2015.
[2] A Krizhevsky, I Sutskever, and G E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS) 25, pages 1097–1105, 2012.
[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.
[4] R Girshick, J Donahue, and T Darrell. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.
[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS) 28, pages 91–99, 2015.
[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In International Conference on Learning Representation (ICLR), 2015.
[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3431–3440, 2015.
[8] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre Antoine Manzagol. Stacked denoising autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research, 11:3371–3408, December 2010.
[9] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2392–2399, 2012.
[10] Yi-Qing Wang and Jean-Michel Morel. Can a Single Image Denoising Neural Network Handle All Levels of Gaussian Noise? IEEE Signal Processing Letters, 21(9):1150–1153, May 2014.
[11] 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, May 2017.
[12] Stamatios Lefkimmiatis. Non-local Color Image Denoising with Convolutional Neural Networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5882–5891, 2017.
[13] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In Advances in Neural Information Processing Systems (NIPS) 25, pages 341–349, 2012.

[14] Xiao Jiao Mao, Chunhua Shen, and Yu Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In Advances in Neural Information Processing Systems (NIPS) 29, pages 2810–2818, 2016.

[15] Ruohan Gao and Kristen Grauman. On-demand learning for deep image restoration. In IEEE International Conference on Computer Vision (ICCV), 2017.

[16] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning Deep CNN Denoiser Prior for Image Restoration. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3929–3938, 2017.

[17] 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, August 2007.

[18] Minchao Ye, Yuntao Qian, and Jun Zhou. Multitask Sparse Nonnegative Matrix Factorization for Joint Spectral–Spatial Hyperspectral Imagery Denoising. IEEE Transactions on Geoscience and Remote Sensing, 53(5):2621–2639, December 2014.

[19] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-Dose CT With a Residual Encoder-Decoder Convolutional Neural Network. IEEE Transactions on Medical Imaging, 36(12):2524–2535, November 2017.

[20] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. Medical Physics, 44(10):e360–e375, October 2017.

[21] C M Stein. Estimation of the mean of a multivariate normal distribution. The Annals of Statistics, 9(6):1135–1151, November 1981.

[22] A Buades, B Coll, and J M Morel. A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation, 4(2):490–530, January 2005.

[23] J Salmon. On Two Parameters for Denoising With Non-Local Means. IEEE Signal Processing Letters, 17(3):269–272, March 2010.

[24] D Van De Ville and M Kocher. SURE-Based Non-Local Means. IEEE Signal Processing Letters, 16(11):973–976, November 2009.

[25] S Ramani, T Blu, and M Unser. Monte-Carlo Sure: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms. IEEE Transactions on Image Processing, 17(9):1540–1554, August 2008.

[26] Charles-Alban Deledalle, Samuel Vaiter, Jalal Fadili, and Gabriel Peyré. Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. SIAM Journal on Imaging Sciences, 7(4):2448–2487, 2014.

[27] T Blu and F Luisier. The SURE-LET Approach to Image Denoising. IEEE Transactions on Image Processing, 16(11):2778–2786, October 2007.

[28] Léon Bottou. Online Learning and Stochastic Approximations. In On-line learning in neural networks, pages 9–42. Cambridge University Press New York, NY, USA, 1998.

[29] Yuri Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In Soviet Mathematics Doklady, 1983.

[30] Diederik P Kingma and Jimmy Ba. Adam - A Method for Stochastic Optimization. In International Conference on Learning Representation (ICLR), 2015.

[31] Martín Abadi et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, pages 265–283, 2016.

[32] Y C Eldar. Generalized SURE for Exponential Families: Applications to Regularization. IEEE Transactions on Signal Processing, 57(2):471–481, January 2009.