Self-Guided Network for Fast Image Denoising

Shuhang Gu[1], Yawei Li[1], Luc Van Gool[1,2], Radu Timofte[1]
[1]Computer Vision Lab, ETH Zurich, Switzerland, [2]KU Leuven, Belgium
{yawei.li, shuhang.gu, vangool, radu.timofte}@vision.ee.ethz.ch

Abstract

During the past years, tremendous advances in image restoration tasks have been achieved using highly complex neural networks. Despite their good restoration performance, the heavy computational burden hinders the deployment of these networks on constrained devices, e.g. *smart phones and consumer electronic products. To tackle this problem, we propose a self-guided network (SGN), which adopts a top-down self-guidance architecture to better exploit image multi-scale information. SGN directly generates multi-resolution inputs with the shuffling operation. Large-scale contextual information extracted at low resolution is gradually propagated into the higher resolution sub-networks to guide the feature extraction processes at these scales. Such a self-guidance strategy enables SGN to efficiently incorporate multi-scale information and extract good local features to recover noisy images. We validate the effectiveness of SGN through extensive experiments. The experimental results demonstrate that SGN greatly improves the memory and runtime efficiency over state-of-the-art efficient methods, without trading off PSNR accuracy.*

1. Introduction

Image denoising is one of the fundamental problems in the signal processing and computer vision communities. Given a noisy observation $\mathbf{y} = \mathbf{x}+\mathbf{v}$, image denoising aims to remove the noise $\mathbf{v}$ and estimate the latent clean image $\mathbf{x}$. With the wide availability of various consumer cameras, the demand for highly accurate and efficient image denoising algorithms has become stronger than ever.

During the past several years, deep neural networks have been very successful at image denoising [10, 1]. By stacking convolution, batch normalization [16], ReLU layers and adopting the idea of residual learning, Zhang *et al*. [45] proposed the DnCNN approach, which achieves a much higher PSNR index than conventional state-of-the-art approaches [7, 11]. The great success achieved by DnCNN has inspired multiple follow-up works. For the pursuit of highly accurate denoising results, some complex networks [25, 38] have been proposed. Although these networks can obtain very competitive denoising performance on benchmark datasets, their heavy computation and memory footprint hinder their application on hardware constrained devices, such as smart phones or consumer electronic products. In Fig. 1, we compare the average running time, peak GPU memory consumption as well as denoising performance by different denoising algorithms on the benchmark dataset *Set68* [26]. As can be seen in the figure, it takes the state-of-the-art MemNet [38] more than 150 ms to process a 480×320 image, which obviously cannot fulfill the requirements of current real-time systems.

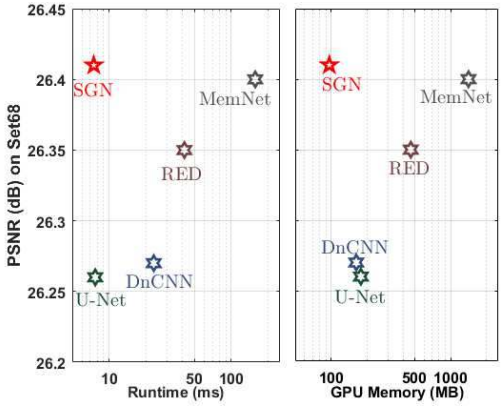


Figure 1. Comparison of PSNR, runtime and peak GPU memory consumption by the proposed SGN ($L3g3m2$) [1] and state-of-the-art approaches DnCNN [45], U-Net [33], RED [25] and MemNet [38]. The runtimes are evaluated on a TITAN Xp GPU.

In this paper, we propose a self-guided neural network (SGN) to seek a better trade-off between denoising performance and the consumption of computational resources. In Fig. 2, we present an SGN with 3 shuffle levels. To better capture the relationship between input and target images, we adopt a top-down guidance strategy to design the network structure of SGN. Concretely, we adopt shuffling operations to generate multi-resolution inputs. Having the multi-resolution input variations, SGN firstly processes the top-branch which is 8× smaller than the original input im-

[1]More details on the memory consumption and PSNR of the proposed SGN with other hyper parameters are provided in the experimental section.

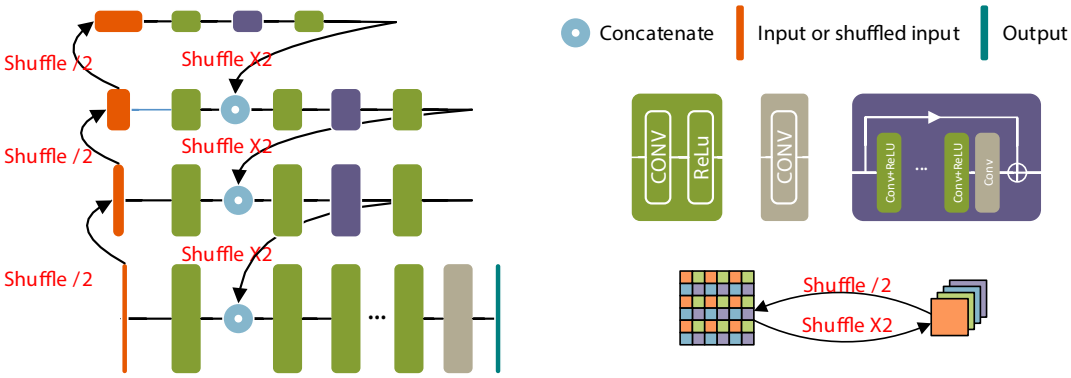


Figure 2. An illustration of Self-guided Network (SGN). SGN directly take shuffled images as multi-resolution inputs, and adopts a top-down self-guidance strategy to better utilizing contextual information. More details of SGN can be found in Sec. 3.

age. Conducting convolutions at such low spatial resolution enables SGN to enlarge its receptive field rapidly. Thus, with only several convolution layers, the top sub-network is able to have an overview of the image content. Then, the contextual information is propagated to a higher resolution sub-network to guide the feature extraction process at the higher resolution branch. Note that we introduce the features from low resolution sub-network, which contain large-scale information, into the high resolution branch as early as possible. Our experiments have show that having a macro-understanding of image structures at an early stage helps the high resolution sub-network to better extract local features, and consequently, leads to better denoising results.

The idea of utilizing multi-scale contextual information to handle dense estimation tasks has been investigated in some previous works [33, 47, 5, 48]. But the proposed SGN is significantly different from previous methods in the following two aspects: a top-down self-guided architecture and shuffled multi-scale inputs. 1) *Top-down self-guidance*: instead of gradually generating multi-scale feature maps in the intermediate layers [33], SGN generates input variations and directly works on the multi-resolution inputs. Furthermore, SGN firstly extracts features in the low resolution branch and propagates contextual information into high resolution branches. 2) *Shuffled multi-scale inputs*: SGN generates multi-resolution inputs with the shuffling operation to avoid information loss introduced by the down-sampling/pooling operation. The shuffling operation has been adopted in some previous image restoration works [46, 34, 37] to change the spatial resolution of image/feature maps. However, to the best of our knowledge, we are the first to directly generate multi-scale inputs via shuffling instead of the down-sampling operation. The advantages of the above strategies are validated with detailed ablation studies.

Our main contributions are summarized as follows:

- A fast and memory efficient network, namely SGN, is proposed to deal with the image denoising task. By adopting a top-down self-guided architecture, SGN outperforms current algorithms in terms of denoising performance, speed and memory efficiency.
- We provide detailed ablation studies to analyze and validate the advantages of the proposed self-guidance strategy.
- Quantitative and qualitative experimental results on synthetic and real datasets have been provided to compare SGN with state-of-the-art algorithms.

2. Related Work

In this section, we review works related to our research. First, we review some DNN-based denoising algorithms and guided image enhancement algorithms. Then, we discuss some previous works for enlargement of the receptive field and for incorporation of multi-scale information.

2.1. Deep Neural Networks for Image Denoising

As one of the most classical image processing problem, image denosing has been intensively studied for many years [10]. One of the earliest attempts to apply convolutional neural networks (CNNs) for image denoising is [17], in which Jain and Seung claimed that CNNs have similar or even better representation power than the Markov random field model. More recently, Xie *et al*. [42] stacked sparse denoising auto-encoders and achieved comparable denoising performance with the K-SVD [3] algorithm. Schuler *et al*. [36] trained a multi-layer perceptron (MLP) for image denoising. The MLP method is the first network to achieve comparable denoising performance with the baseline BM3D [7] approach. After MLP, Schmidt *et al*. [35] and Chen *et al*. [6] unfolded the inference process of the optimization-based denoising model to design denoising networks. Zhang *et al*. [45] stacked convolution, batch normalization [16] and ReLU [27] layers to estimate the residual between the noisy input and the corresponding clean image. Inspired by the success of DnCNN, Mao *et al*. [25]

proposed a very deep residual encoding-decoding (RED) framework to solve image restoration problems, in which skip connections have been introduced to train the very deep network. Tai *et al*. [38] proposed a very deep persistent memory network (MemNet) for image denoising. Recursive unit and gate unit have been adopted to learn multi-level representations under different receptive fields. In addition, Liu *et al*. [22] incorporated non-local operations into a recurrent neural network (RNN) and proposed a non-local recurrent network (NLRN) for image restoration. Liu *et al*. [23] modified U-Net [33] architecture, and proposed a multi-level wavelet CNN (MWCNN) model to incorporate large receptive field for image denoising. The NLRN [22] and MWCNN methods [23] improve the denoising performance over DnCNN [45], but also have a higher demand on computational resources.

2.2. Guided Image Restoration

The idea of utilizing guidance information for improving image restoration performance has been validated in many previous works. Bilateral filter [39], guided filter [14] and their variations [19, 29] utilize an external image to adjust local filter parameters. They achieved very good performance on a wide range of low-level vision tasks, including flashing and no-flashing photography [32], image matting [14], depth upsampling [29], etc. By incorporating both the depth map and RGB image into a joint objective function, guided depth super-resolution approaches [8, 9, 13] achieved much better performance than the plain depth super-resolution methods. In the last several years, guidance information has also been introduced in DNN-based models to pursuit better image restoration performance. In [21, 15, 12], features extracted from an aligned RGB image have been exploited for guiding the super-resolution of depth map. In [20], a high quality face image has been utilized to guide the enhancement of low quality face images from the same person. Recently, Wang *et al*. [41] proposed to use the semantic segmentation map to guide the super-resolution of input image. By introducing high-level semantic information at an early stage of super-resolution network, [41] generates more realistic textures in the super-resolution results. The success of the above methods indicate that appropriate guidance information is beneficial for image restoration. In this paper, instead of incorporating external information, SGN adopts a self-guidance strategy. Multi-resolution inputs are generated by the shuffling operation, we extract large-scale information at low-resolution branch to guide the restoration process at fine scale.

2.3. Receptive Field Enlargement

Large receptive field is critical to the learning capacity of CNN. In contrast to high-level problems such as classification and detection, which can obtain large receptive field by successively down-sampling the feature map with pooling or strided convolution, dense estimation tasks need to predict labels for each pixel in the image. Thus, how to incorporate contextual information from a large receptive field on a full resolution output is a challenging problem. In this sub-section, we present some related works proposing receptive field enlargement for dense estimation tasks.

One line of approaches utilizes dilated convolution [43] to increase the receptive field of DNN. After the seminal work by Yu *et al*. [43], a large number of recently proposed semantic segmentation works [48, 40] have adopted dilated convolution for incorporating information from large surrounding area. In the field of image manipulation, Chen *et al*. [5] proposed a context aggregation network (CAN) to incorporate multi-scale contextual information. Another category of approaches incorporates contextual information by down-sampling and up-sampling the feature maps in the middle of networks. The encoder-decoder structure has been adopted in [31, 44] for the purpose of incorporating image global information. U-shape networks [33, 23, 28] method uses successive pooling approach to gradually reduce the spatial resolution of feature maps and use up-convolution operation to recover the feature map back to the original resolution. Our work share a similar idea of extracting information at low resolution to enlarge the receptive field of network. However, different from previous works [33] which gradually reduce the spatial resolution of feature maps, SGN starts to process the input at the lowest-resolution branch which is directly generated with the shuffling operation. Our ablation experiments show that the earlier large-scale information is incorporated, the better is the restoration performance that can be achieved.

3. Self-Guided Network (SGN)

In this section, we introduce the network structure of the proposed SGN. We firstly present the overall network structure of SGN. Then, we introduce details of each sub-network and discuss the setting of hyper parameters.

3.1. Overall structure of SGN

The core idea of this work is utilizing large-scale contextual information to guide the image restoration process at finer scales. Given an input image $\mathbf{I}^0$ with dimension $M \times N \times C$, SGN firstly shuffles $\mathbf{I}^0$ to a series of variations $\mathbf{I}^k$ with spatial dimension $\{M/2^k \times N/2^k \times 4^k C\}_{k=1,\dots,K}$. Then, top sub-network $f^K(\cdot)$ at level K firstly extracts features from $\mathbf{I}^K$. Since the spatial resolution of $\mathbf{I}^K$ is 2^K times smaller than that of the original input, conducting convolution at sub-network $f^K(\cdot)$ increases the network perceptive field 2^K times faster than conducting convolution at the full resolution branch $f^0(\cdot)$. Thus, $f^K(\cdot)$ is able to extract large-scale contextual information efficiently. Having the large scale information, we propagate it to the higher

resolution branch $f^{K-1}(\cdot)$ to guide the feature extraction process at that scale. Specifically, the guidance information will be introduced at the beginning of the sub-networks to help each sub-network have an overall understanding of the context at an early stage. Through middle sub-network $\{f^k(\dot{)}\}_{k=1,...,K-1}$, the multi-scale contextual information gradually moves to the full resolution and guides the bottom sub-network $f^0(\cdot)$ to generate the final estimation.

3.2. Network architecture details

SGN consists of three kinds of sub-networks: 1) *top* sub-network $f^K(\cdot)$ at lowest spatial resolution, 2) *middle* sub-networks $\{f^k(\cdot)\}_{k=1,...,K-1}$ at intermediate resolution 3) *bottom* sub-networks $f^0(\cdot)$ at the full resolution branch.

3.3. Top sub-network

The top sub-network works on a very low spatial resolution to extract large scale information. As shown in Fig. 2, the top sub-network contains two Conv+ReLU layers and a residual block. Because the target of this paper is to design a fast denoising algorithm, we adopt lightweight structures for both the top, middle and bottom sub-networks. Thus, we do not introduce many small residual blocks and only use one skip connection to form a residual block in each sub-networks. By varying the number of convolutions in the residual block, we obtain different operating points for SGN. We denote the number of convolution layers between the skip connection as g. In Fig. 2, the blue box represents a residual block with $g = 3$.

The shuffling operation reduces spatial resolution of $\mathbf{I}^0$, but increases its channels. After shuffling, the channel number of $\mathbf{I}^K$ is 4^K times larger than the channel number of the original input $\mathbf{I}^0$. We adopt more feature maps at the top sub-network to better extract features. Denote the number of feature maps at full resolution sub-network $f^0(\cdot)$ as c^0, for the top sub-network $f^K(\cdot)$ we set the number of feature maps as $c^K = 2^K c^0$. Note that because the spatial size of feature maps at top sub-network is $\times 4^K$ smaller than the full resolution feature maps, conducting convolution at level K is still much faster than conducting it at full resolution.

3.3.1 Middle sub-network

The network structure of the middle sub-networks is similar with the structure of the top sub-network. The number of feature maps $f^k(\cdot)$ is set as $2^k c^0$, and the residual block used in the middle sub-network also contains g convolution layers. The only difference is that middle sub-networks need to incorporate guidance information from its upper sub-network. Specifically, we adopt a *shuffle* $\times 2$ operation to enlarge the spatial resolution of feature maps extracted from $f^k(\cdot)$ and concatenate them with the output of first layer in $f^{k-1}(\cdot)$. As the concatenate operation increases the number of feature maps from c^k to $\left(c^k + c^{k+1}/4\right)$, we adopt an extra convolution layer in middle sub-networks to reduce the number of feature maps back to c^k.

Please note that other guidance incorporating methods, such as feature map multiplication [13] or the sophisticated Spatial Feature Transform (SFT) block [41], can also be utilized to introduce the guidance information. But as the main goal of this paper is to present the overall framework of SGN architecture, here we utilize a simple concatenate operation which is the most commonly used operation for fusing feature maps. The simple concatenate operation also helps us to achieve a good trade-off between denoising accuracy and speed.

3.3.2 Bottom sub-network

The bottom sub-network utilizes the same guidance incorporation method as the middle sub-network. The shuffled guidance feature maps from $f^1(\cdot)$ are concatenated with the output of the first convolution layer in bottom sub-network, then, we adopt a convolution layer to reduce the feature map number from $\left(c^0 + c^1/4\right)$ back to c^0. For the denoising task, as we have a global residual connection between the input image and the final estimation, we do not use any residual blocks in the bottom sub-network. After the guidance block, we use m Conv + ReLU layers to further process the joint feature map, and the final estimation is generated with an extra convolution layer. The bottom sub-network contains $(m + 3)$ convolution layers in total.

3.4. SGN Parameters

SGN has hyper-parameters g, m, c^0 and K. Hyper-parameter g controls the depth of top and middle sub-networks, m controls the depth of bottom sub-network, c^0 is the feature map number, and K is the number of shuffling levels in SGN.

In this paper, we set the network depth parameters $g = 3$, $m = 2$ and $c^0 = 32$ to achieve a balance between performance and efficiency. The top, middle and bottom sub-networks contain 5, 6 and 5 convolution layers, resp. The level parameter K affects the spatial resolution of top sub-network. For synthetic Gaussian denoising task, we set $K = 3$ to compare SGN with other denoising algorithms. While, for the more challenging noisy raw to image dataset [4], we adopt $K = 4$ to achieve larger receptive field. Experimental results as well as discussion towards K will be provided in 4. Our source code and more experimental results by SGN with other hyper-parameters can be found in our project webpage[2].

4. Experiments: Ablation Study

In this section, we conduct experiments to validate the effectiveness of the proposed SGN network. We firstly in-

[2]https://github.com/ShuhangGu/SGN_ICCV2019

troduce experimental settings and then validate the advantages of top-down self-guided structure as well as the shuffling operation for multi-scale inputs generation.

4.1. Experimental Setting

In the ablation study section, we evaluate the proposed method on gray image denoising task. In order to thoroughly evaluate the capacity of network, we use the 800 high resolution ($2040 \times 1550+$) training images provided in the DIV2K dataset [2] as training set, and use the 100 images validation set of DIV2K as testing set. To compare with previous denoising algorithms, the denoising results on the commonly used 68 images from Berkeley segmentation dataset [26] are also provided for reference. We follow the experimental setting in MemNet [38] and evaluate denoising methods with additive white Gaussian noise with standard variation σ of 30, 50 and 70. All the methods have been evaluated on the same noisy samples (noise generated with Matlab random seed 0).

As different training data has been adopted in previous methods [45, 38, 25], for the purpose of fair comparison, we retrain all the competing approaches on our training dataset. All the competing approaches as well as the proposed SGN are implemented with the Pytorch toolbox [30]. The network is trained with the Adam [18] solver with parameters $\beta_1 = 0.9$. As SGN has a large receptive field, in each iteration, we randomly crop 8 sub-images with size 256×256 from the training set. Online data augmentation with random flip and rotation operations have been adopted to further increase the training data. We train our model with learning rate 1×10^{-4} for 500K iterations and then reduce the learning rate to 1×10^{-5} for another 500K iterations.

4.2. Effectiveness of multi-scale information

We firstly show that multi-scale processing could significantly improve denoising performance. To show the effectiveness of multi-scale information, we evaluate the denoising results by SGN with different number of levels. SGN with level 0, denoted SGN_{L0}, uses only the bottom sub-network. Similarly, SGN with level K is denoted as SGN_{LK}. SGN_{L2}, SGN_{L3} and SGN_{L4} are normal SGN networks, while SGN_{L1} directly uses the information from top sub-network to guide the denoising process at full resolution. The denoising results by SGN networks with different level numbers are shown in Table 1. The GPU memory consumption as well as the running time for processing a 320×480 image is also provided for reference.

The results clearly show the advantage of incorporating multi-scale information for denoising. The denoising results improve with the number of levels in SGN. Furthermore, by utilizing only 6 more convolution layers at small spatial resolution, SGN_{L1} improves the performance of SGN_{L0} on DIV2K for more than 1 dB.

Table 1. Denoising results (PSNR) and computational consumption (peak GPU memory, runtime) by SGN with different level numbers. Noise level $\sigma = 50$.

Method	PSNR [dB]		GPU [GB]	Time [ms]
	BSD68	DIV2K		
SGN_{L0}	25.48	26.95	0.0654	3.1
SGN_{L1}	26.15	28.03	0.0746	4.6
SGN_{L2}	26.36	28.42	0.0817	6.2
SGN_{L3}	26.46	28.57	0.0942	7.5
SGN_{L4}	26.48	28.58	0.1324	9.1

4.3. Self-guided feature extraction

As we have discussed in the previous sections, the core idea of this work is utilizing large-scale contextual information to guide the image restoration process at finer scales. In this part, we validate the idea of guided feature extraction. We conduct experiments on a $K = 1$ network, and show that the earlier we incorporate contextual information at fine-scale the better denoising performance we could achieve. We train the three networks in Fig. 3 for image denoising, which incorporate contextual information to the full resolution at different stages. Specifically, the first network is the proposed SGN ($g = 3, m = 4$). It introduces the large-scale information at the beginning of fine-scale branch; while, the remaining two networks introduce contextual information in the middle or the end of fine scale sub-network. We denote the three networks as SGN_{early}, SGN_{middle} and SGN_{late}, resp. Their denoising results can be found in Table 2. On both the Set68 and DIV2K 100 dataset, the proposed SGN (SGN_{early}) achieved the best denoising performance. Furthermore, the results clearly shows that the earlier we introduce the contextual information, the better performance we can achieve. The results in Table 2 demonstrate the advantage of our top-down self-guidance strategy, and we think it is the major reason that the proposed SGN outperform previous multi-scale methods such as U-Net [33].

Table 2. Denoising results (PSNR [dB], $\sigma = 50$) by SGN_{early}, SGN_{middle} and SGN_{late} presented in Fig. 3.

Dataset	SGN_{early}	SGN_{middle}	SGN_{late}
Set68	26.16	26.12	26.06
DIV2K	28.08	28.02	27.92

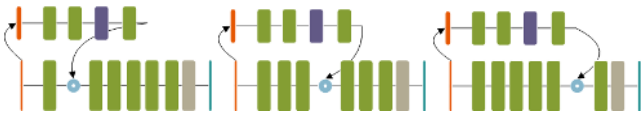


Figure 3. A one-level SGN with early (*left*), middle (*middle*) and late (*right*) guidance. The denoising results by the three networks can be found in Table 2.

4.4. Shuffling *vs.* Down-Sampling

In order to extract large-scale contextual information at an early stage, SGN shuffles the original image and generates its multi-scale variations as inputs. In some previous semantic segmentation works [47], multi-scale variations of input image have also been adopted directly as inputs of networks. However, as semantic segmentation task does not need to infer fine texture details, the multi-scale variations are often achieved by down-sampling the original input. For the image denoising task, as fine details are very important, we use shuffling instead of down-sampling operation to generate multi-scale inputs. Compared with down-sampling, shuffling is able to reduce spatial resolution but keeps all information of input image. As a result, each sub-network in SGN could adaptively extract better features at different scales for image denoising. In Table 3, we compare the standard SGN and its variation which take down-sampled input images as inputs to different sub-networks. The network with shuffled inputs achieves better performance than the competing method with down-sampled inputs. Note that the shuffling operation has been adopted in some previous image restoration works [46, 34, 37] to change the spatial resolution of image/feature maps. Our method is the first work which utilizes shuffling operation to generate multi-scale variations of input image. The shuffling operation and our self-guided strategy cooperated to deliver our full SGN algorithm.

Table 3. Shuffle/Down-sample for generating multi-scale inputs in SGN.

Dataset	Set68	DIV2K
Shuffling→down-sampling	26.43→26.20	28.53→28.18

5. Experiments: Gaussian Noise Removal

In this section, we compare SGN with state-of-the-art denoising approaches. We first introduce the competing approaches briefly and then provide experimental results on both the gray-scale and RGB image denoising tasks.

5.1. Compared Approaches

The compared approaches include commonly used denoising algorithms DnCNN [45], RED [25] and MemNet [38]. A comparison of our more powerful SGN (incorporating more convolutional layers) with latest denoising algorithms [22, 23] can be found in our project webpage. We also provide the denoising results by related works U-Net [33] and CAN [5]. Both U-Net and CAN proposed to incorporate large scale contextual information for better image manipulation. CAN [5] utilizes dilated convolution and U-Net [33] is a representative approach of extracting feature at different spatial resolution. For both CAN [5] and U-Net [33], we use the same number of feature maps as the proposed SGN, *i.e*. 32 feature maps at full resolution. We follow the authors of U-Net [33] and process images at 5 different scales. For CAN [5] approach, we follow the setting in the original paper and utilize 9 dilated convolution layers with different dilation parameters.

Our SGN model is trained on the training dataset of DIV2K [2] which contains 800 high resolution images. For the purpose of fair comparison, we implement all the networks by ourselves and retrain all the methods on the same training dataset as we used.

We have tried our best to train the competing methods. Concretely, we use the same training parameters described in Sec. 4.1 to train CAN [5], U-Net [33], RED [25] as well as the proposed SGN. Since DnCNN [45] and MemNet [38] approaches adopt batch normalization [16] layers in their networks, they require a large batch size for good performance. We follow the batch size setting of the original authors and set batch size for DnCNN [45] and MemNet [38] to 64. Due to memory limitation, we are not able to train DnCNN [45] and MemNet [38] with sub-images of size 256×256. We set the training patch size for DnCNN [45] and MemNet [38] as 64×64 and 48×48, respectively. The patch sizes adopted for both methods are larger than the value adopted in the original papers.

5.1.1 Gray Image Denoising

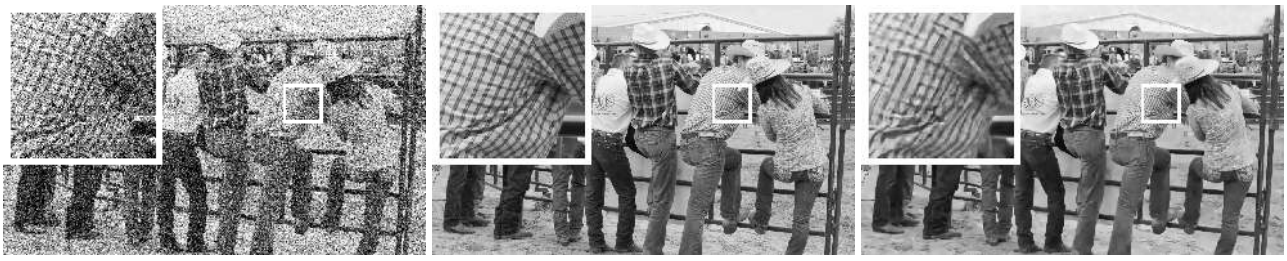
We firstly compare different algorithms on the gray-scale image denoising task. The image denoising results by different methods on the Set 68 and DIV2K validation 100 dataset are shown in Table 4. The running time and GPU memory consumption for processing a 320×480 image with noise level $\sigma = 50$ are shown in Table 5.

Generally, the proposed SGN outperforms the competing methods on all the noise levels. Taking computational burden into consideration, the proposed SGN shows great advantages over the competing approaches. All the competing methods require more GPU memory and have longer running time than the proposed SGN_{L3} network. Compared with the state-of-the-art MemNet [38], SGN is not only about 20 times faster but also consuming 15 times less GPU memory to denoise one image.

A visual example of the denoising results by different methods are shown in Fig. 4.

5.1.2 Color Image Denoising

We also evaluate different algorithms on the color image denoising task. We use the color version of DIV2K training dataset to train all the models. The setting of training parameters for different methods are the same as our experiments for gray image denoising. The PSNR value by different methods are shown in Table 6. The proposed SGN outperforms all the competing approaches.



Noisy Input | Ground Truth | DnCNN [45]

RED [25] | MemNet [38] | SGN (Ours)

Figure 4. Denoising results by different methods on a testing image from the DIV2K dataset ($\sigma = 70$).

Table 4. Gray image denoising results (PSNR) by different methods.

Dataset	Noise Level	CAN [5]	U-Net [33]	DnCNN [45]	RED [25]	MemNet [38]	SGN$_{L3}$
BSD 68	$\sigma = 30$	26.28	28.29	28.43	28.46	28.46	**28.50**
	$\sigma = 50$	24.82	26.26	26.30	26.35	26.40	**26.43**
	$\sigma = 70$	24.17	25.03	25.00	25.05	24.99	**25.17**
DIV2K 100	$\sigma = 30$	27.28	30.48	30.55	30.59	30.51	**30.71**
	$\sigma = 50$	24.99	28.34	28.25	28.39	28.50	**28.53**
	$\sigma = 70$	25.37	27.03	26.79	26.92	26.86	**27.10**

Table 5. Peak GPU memory consumption [GB] and time usage [ms] by different methods for process a 480×320 image. All the methods were implemented under PyTorch [30], the running time is evaluated on an Nvidia Titan Xp GPU.

Methods	CAN [5]	U-Net [33]	DnCNN [45]	RED [25]	MemNet [38]	SGN$_{L3}$	SGN$_{L4}$
GPU consumption [GB]	0.1199	0.1731	0.1583	0.4530	1.3777	0.0942	0.1323
Time [ms]	10.1	7.7	23.2	41.3	156.8	7.5	9.1

Table 6. RGB image denoising results (PSNR [dB]).

Dataset	Noise Level	U-Net [33]	DnCNN [45]	RED [25]	MemNet [38]	SGN$_{L3}$ (ours)
BSD 68	$\sigma = 30$	30.30	30.31	30.40	**30.45**	**30.45**
	$\sigma = 50$	28.03	28.03	28.04	28.08	**28.18**
	$\sigma = 70$	26.69	26.50	26.62	26.59	**26.79**
DIV2K 100	$\sigma = 30$	31.93	31.99	32.14	32.20	**32.21**
	$\sigma = 50$	29.74	29.79	29.82	29.85	**30.02**
	$\sigma = 70$	28.40	28.13	28.33	28.37	**28.62**

6. Enhancement of Image Raw Data

In this section, we validate the proposed SGN on a more challenging See-in-the-Dark (SID) dataset [4]. The SID dataset is collected by Chen *et al*. to support the development of learning based pipelines for low-light image processing. 5094 raw short-exposure images with corresponding long-exposure reference images have been provided in the dataset. The short exposure images were captures in extreme low-light conditions with two cameras: Sony α7S II and Fujifilm X-T2. Neural network needs to learn the image processing pipeline for low-light raw data, including color transformations, demosaicing, noise reduction, and image enhancement.

In [4], U-Net [33] with LeakyReLU has been suggested to build the mapping function from low-light raw data to high quality images. In this paper, we follow the experi-

Reference Image U-Net [33] SGN_{L3} SGN_{L4}

Figure 5. Results by U-Net and SGN_{L3}, SGN_{L4} on an image from the SID [4] dataset.

U-Net [33, 4] Reference Image SGN_{L4}

Figure 6. Zoomed results by U-Net and SGN_{L4} on SID images [4].

mental setting as [4]. We utilize the same raw-data preprocessing scheme and train SGN with the same training data provided by [4]. As [4] shows that the L_1 loss leads to better mapping accuracy than the L_2 loss, we thus train SGN on SID dataset with the L_1 loss. In addition, we also change our activation function to LeakyReLU [24] as [4].

In the SID dataset, the network needs to capture the color transformation between the input data and output image. Thus, large-contextual information is more important than in the Gaussian denoising task. We report the estimation results by both the SGN_{L3} and SGN_{L4}. The estimation results by the proposed SGN_{L3}, SGN_{L4} as well as the competing methods CAN [5] and U-Net [33] are shown in Tab. 7. The PSNR values by the CAN [5] and U-Net [33] method are provided by [4]. On both the Sony and Fuji sub-datasets, the proposed SGN achieved better performance than the U-Net [33]. On the more challenging Fuji sub-dataset, SGN_{L4} achieved a 0.8 dB higher PSNR than U-Net [33].

In Fig. 5, we present a visual example of the estimation results by U-Net [33] and SGN. U-Net severely changed the color on the wall, while both the SGN_{L3} and SGN_{L4} generated high quality estimations. Note that the estimation of the correct color relies on incorporating large-scale contextual information. As U-Net adopts 4 pooling layers and its low-resolution branch is 2 times smaller than the low-resolution branch of SGN_{L3}, its perceptive field actually is larger than the perceptive field of SGN_{L3}. Attributed to the self-guided strategy, SGN_{L3} is able to incorporate contextual information at an early stage, and thus better estimates the color in the output.

Fig. 6 shows the advantage of SGN over U-Net [33] in recovering image details for two examples.

Table 7. Quantitative comparison in terms of PSNR [dB] between SGN and CAN [5], U-Net [33], and our SGN on the SID dataset.

Methods	CAN [5]	U-Net [33]	SGN_{L_3}	SGN_{L_4}
Sony Dataset	27.40	28.88	28.91	**29.06**
Fuji Dataset	25.71	26.61	26.90	**27.41**

7. Conclusion

In this paper, we proposed a self-guided neural network (SGN) for fast image denoising. SGN adopts a self-guidance strategy to denoise an image in a top-down manner. Given the input image, shuffling operations are adopted to generate input variations with different spatial resolutions. Then, SGN extracts feature at low spatial resolution and utilizes the large scale contextual information to guided the feature extraction process at finer scales. The multi-scale contextual information gradually moves back to the full resolution branch to guide the estimation of output image. The proposed SGN was validated on standard gray image and color image denoising benchmarks. Our SGN is able to generate high quality denoising results with much less running time and GPU memory consumption than the compared state-of-the-art methods.

Acknowledgments: This work was supported by Huawei, the ETH Zurich General Fund, and an Nvidia GPU hardware grant.

References

[1] Abdelrahman Abdelhamed, Radu Timofte, and Michael S. Brown. Ntire 2019 challenge on real image denoising: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, 2017. 5, 6

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 2006. 2

[4] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018. 4, 7, 8

[5] Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. 2, 3, 6, 7, 8

[6] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 2

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 2007. 1, 2

[8] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *Neural Information Processing Systems*, 2005. 3

[9] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *IEEE International Conference on Computer Vision*, 2013. 3

[10] Shuhang Gu and Radu Timofte. A brief review of image denoising algorithms and beyond. 2019. 1, 2

[11] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 1

[12] Shuhang Gu, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, and Lei Zhang. Learning dynamic guidance for depth image enhancement. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2017. 3

[13] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust image filtering using joint static and dynamic guidance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4823–4831, 2015. 3, 4

[14] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *European conference on computer vision*, pages 1–14. Springer, 2010. 3

[15] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *European Conference on Computer Vision*, pages 353–369. Springer, 2016. 3

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1, 2, 6

[17] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009. 2

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

[19] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (ToG)*, 26(3):96, 2007. 3

[20] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. *arXiv preprint arXiv:1804.04829*, 2018. 3

[21] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *European Conference on Computer Vision*, 2016. 3

[22] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018. 3, 6

[23] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 773–782, 2018. 3, 6

[24] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 8

[25] Xiaojiao 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*, pages 2802–2810, 2016. 1, 2, 5, 6, 7

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 1, 5

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 2

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 3

[29] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S Brown, and Inso Kweon. High quality depth map upsampling for 3d-tof cameras. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1623–1630. IEEE, 2011. 3

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5, 7

[31] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 3

[32] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. In *ACM transactions on graphics (TOG)*, volume 23, pages 664–672. ACM, 2004. 3

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 2, 3, 5, 6, 7, 8

[34] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-Recurrent Video Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 6

[35] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *CVPR*, 2014. 2

[36] Christian J Schuler, Harold Christopher Burger, Stefan Harmeling, and Bernhard Scholkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, 2013. 2

[37] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 2, 6

[38] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *CVPR*, 2017. 1, 3, 5, 6, 7

[39] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998. 3

[40] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460. IEEE, 2018. 3

[41] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. *arXiv preprint arXiv:1804.02815*, 2018. 3, 4

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

[43] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[44] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018. 3

[45] 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*, 2017. 1, 2, 3, 5, 6, 7

[46] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *arXiv preprint arXiv:1710.04026*, 2017. 2, 6

[47] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. *arXiv preprint arXiv:1704.08545*, 2017. 2, 6

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 2, 3