GradNet Image Denoising

Yang Liu[1,2], Saeed Anwar[1,2], Liang Zheng[1], Qi Tian[3]
Australian National University[1], Data61-CSIRO[2], Huawei Noah's Ark Lab[3]
yang.liu3@anu.edu.au, saeed.anwar@csiro.au, liang.zheng@anu.edu.au, tianqi1@huawei.com

Abstract

High-frequency regions like edges compromise the image denoising performance. In traditional hand-crafted systems, image edges/textures were regularly used to restore the frequencies in these regions. However, this practice seems to be left forgotten in the deep learning era. In this paper, we revisit this idea of using the image gradient and introduce the GradNet. Our major contribution is fusing the image gradient in the network. Specifically, the image gradient is computed from the denoised network input and is subsequently concatenated with the feature maps extracted from the shallow layers. In this step, we argue that image gradient shares intrinsically similar nature with features from the shallow layers, and thus that our fusion strategy is superior. One minor contribution in this work is proposing a gradient consistency regularization, which enforces the gradient difference of the denoised image and the clean ground-truth to be minimized. Putting the two techniques together, the proposed GradNet allows us to achieve competitive denoising accuracy on three synthetic datasets and three real-world datasets. We show through ablation studies that the two techniques are indispensable. Moreover, we verify that our system is particularly capable of removing noise from textured regions.

1. Introduction

This paper studies the image denoising problem, which is a basic low-level image processing task that aims to recover a clean image $\boldsymbol{x}$ from its noisy observation $\boldsymbol{y}$. Usually the additive noise model $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ is assumed, where $\boldsymbol{n}$ is the noise. The noise generation process is highly complicated and non-trivial to model, thus making this task challenging. Traditional methods in this community use geometric features of clean images or the noise, such as low rank [18], sparse coding [16, 15], self-similarity [33]. The problem is that these methods may smooth the edge and texture details. In fact, these image details are somehow similar to the noise in that they both encode high frequency (rapid pixel changes). For this problem, many edge and texture preserving denoising methods were proposed in earlier years. Based on the observation that natural image gradients exhibit a heavy-tailed distribution, gradient-based priors are widely used in traditional model-based methods [44]. They usually use a statistical model to estimate image gradient distributions of clean images. However, these methods are compromised by the limitation of the statistical models and the difficulty in image gradient estimation.

(a) Noisy (b) DnCNN (c) GradNet (d) GT Noise

Figure 1: Illustration of our motivation. Given (a) the noisy image, we show the denoising outcome of the bounding box region using DnCNN [40] and GradNet on the top row of (b) and (c), respectively and (d) is the ground truth Gaussian noise. The bottom row of (b) and (c) show the removed noise by DnCNN and GradNet, respectively. While (d) is pure Gaussian noise, we observe the noise in (b) contains some high-frequency residuals of the skirt and arm regions. The contrast is artificially enhanced here for better visualization. This motivates us to fuse image gradient such that the noise in (c) contains much fewer high-frequency residuals, thus better preserving the edge and texture details in the denoised image.

Although deep learning methods achieve superior denoising accuracy [40] [27] [19] [11], we (and several other works [41] [13] [31]) find it difficult for the network to restore edges and fine-scaled textures. An example is shown in Fig. 1 (c), where we observe the shape of the arm and skirt remaining in the estimated noise. That is, pixels in these highly textured regions are treated as noise by the network. As such, we leverage the idea of image gradient from the early days to the deep network, so as to explicitly let edge/texture be differentiated from the noise.

In spite of seemingly straightforward, it is non-trivial to

find an effective way of adopting the image gradient for denoising. To our knowledge, there are very few deep learning works adopting this idea. The first consideration is where in the network we should insert the image gradient. In fact, features represented by the image gradient are of roughly the same significance of those extracted by early layers, and hence directly adding it to the input may not be optimal. Furthermore, late fusion might be even worse, because the network does not have the chance to learn effective representations of the image gradient.

The second consideration is how to acquire the gradient of a clean image . In fact, we can compute image gradients from clean images during training, but cannot obtain such clean gradients in testing. Without deliberate design, testing with noisy image gradients fused would significantly compromise the system.

In the attempt to incorporate image gradient in deep denoising networks, this paper introduces GradNet that takes into account the above considerations and allows us to better preserve the high-frequency texture and edges in the denoised image. Two main solutions are proposed in GradNet. **First**, we fuse image gradient in the shallow network layers instead of the image-level early fusion or late fusion. According to Zeiler and Fergus [39], filters in the shallow layers are sensitive to low-level visual stimulus such as edges and textures. Therefore, the output feature maps from shallow layers share similar physical meanings with the image gradient, which serves as a solid theoretical foundation validating our fusion strategy. **Second**, we use a denoising method to obtain the approximate clean image first and use this denoised image to replace the ground truth image. A gradient consistency loss is proposed as a supplementary. It aims to minimize the difference of image gradient between a denoised image and its possible ground truth clean image, and is used as a regularization of the main task L_1 or L_2 loss. We conduct experiment on a wide range of datasets from synthetic benchmarks to real ones. With GradNet, we report very competitive denoising accuracy compared with the state-of-the-art under an efficient model size.

2. Related Work

Traditional denoising methods with edge and texture preserving properties. Typically, image denoising techniques smooth the noise but also blur important features like edges and corners. Many edge-preserving image denoising methods [20, 30, 38] are introduced after Jain & Tyagi [22] highlighted the importance of edges in a survey. Some of them employed image gradient. Zuo *et al*. [44] enforce the gradient distribution of the denoised image to be close to the estimated gradient distribution of the original image. Komander *et al*. [23] use a variational method to form an optimization problem that aims to denoise the gradient of the noisy image. With this denoised gradient, they minimize its difference with the gradient of the denoised image.

Our method is different from these traditional algorithms not only in terms of using a CNN but also the mechanism employed to incorporate the gradient information. For example, the traditional methods either rely on geometric features of images or employ statistical models of the image gradient. Furthermore, to solve the problem, an optimization scheme is utilized with various priors as penalties or constraints. However, our method learns these patterns implicitly without using any explicit formulation. Similarly, the traditional methods only proved to be effective on synthetic datasets with AWGN, while recent studies show that although the methods targeting AWGN works on synthetic data, their generalization ability in real-world noisy images are limited [41, 19]. Moreover, most of the traditional algorithms require noise as input and various hard thresholds to counter different types of noise levels. On the contrary, our method neither requires noise level nor any hard threshold, and to further demonstrate the effectiveness of GradNet, we train and test on a wide range of synthetic and real noisy datasets and achieve competitive denoising accuracy.

Deep learning for denoising. The research trend of CNN based denoising in recent years can be roughly summarized into three streams. **Firstly**, proposing new real-noise datasets composing of noisy and ground-truth image pairs for evaluation [32] [10] [36] [1] [8] . **Secondly**, adjusting previous state-of-the-art denoising methods that worked for synthetic noise to real noise datasets while considering the gap between these two noise types [41] [19]. **Lastly**, modeling real noise with more complicated distributions and designing new network architectures [7] [11].

However, few attempts are made to incorporate edges and textures in CNN denoising. Recently, a deep convolutional neural network with edge feature (DCEF) for AWGN denoising is presented in [13]. They use Canny edge detector [9] to extract the edges of a noisy image and add the edge map into the noisy image as input. The performance of [13] is only tested on synthetic noisy images while the capacity is limited on real noisy images as the edges are used as an input rather than as features. Pandey *et al*. [31] utilized the ℓ_2 loss between edges of clean images and the denoised images got by Canny operator as a regularization. In this paper, we employ image gradient to extract edge and texture information and explore novel techniques to embed the image gradient into denoising network.

3. Proof of Concept

3.1. Image Gradient Revisit

The image gradient measures the change in pixel intensity on a given direction of an image. By computing the image gradient, we obtain the texture and edge information in an image. There exist some widely used filters to calculate

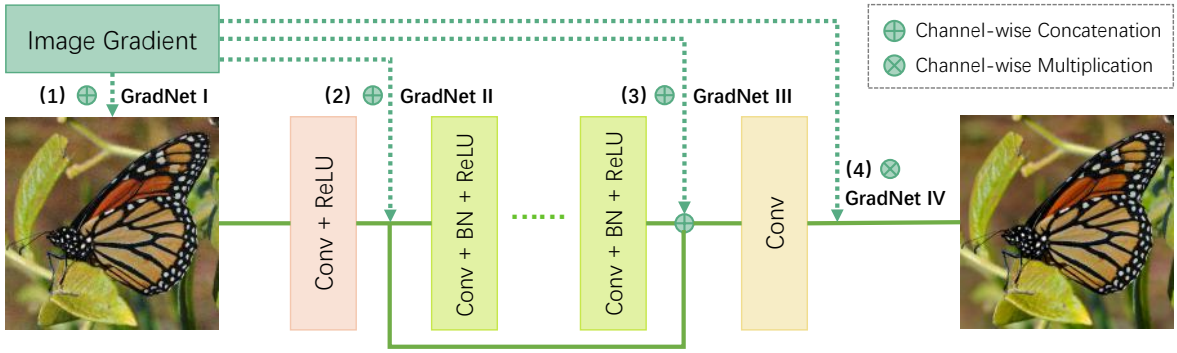


Figure 2: Network variants regarding the fusion of the image gradient. We concatenate the image gradient to (1) the noisy input image, (2) the 64-channel feature maps extracted from the 1st convolutional layer, or (3) the (64+64)-channel feature map obtained after the second last convolutional layer. In (4), we use the image gradient as the weight of the loss in each pixel. The structures (1), (2), (3), and (4) are termed as GradNet-I, GradNet-II, GradNet-III, and GradNet-IV, respectively.

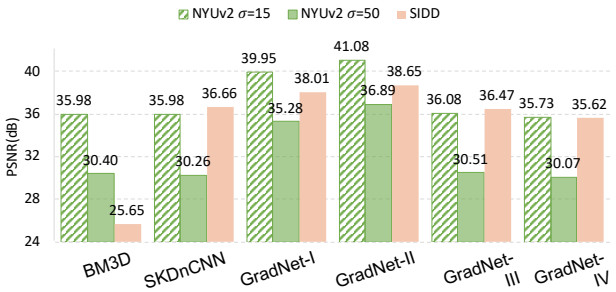


Figure 3: Comparison of different networks under two noise levels and two datasets. GradNet-I and GradNet-II (structure defined in Fig. 2) improve the PSNR over BM3D and SKDnCNN under all settings. The PSNR of GradNet-II is higher than GradNet-I by 1dB and 0.64dB on the synthetic NYUv2 and the real SIDD datasets, respectively. We clear observe GradNet-II has the best denoising performance.

the image gradient, such as Sobel filter [34], Scharr, Prewitt, and Roberts. This paper uses the Sobel filter, composed of a horizontal filter $\boldsymbol{S}_H$ and a vertical filter $\boldsymbol{S}_V$ defined below,

$$S_H = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, S_V = \begin{bmatrix} -1 & -2 & +1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}. \quad (1)$$

We calculate the convolution between $\boldsymbol{S}_H$ or $\boldsymbol{S}_V$ and the image $\boldsymbol{x}$ to obtain the horizontal gradient $\boldsymbol{G}_H$ and vertical gradient $\boldsymbol{G}_V$, respectively. This process can be written as,

$$\boldsymbol{G}_H(\boldsymbol{x}) = \boldsymbol{S}_H * \boldsymbol{x}, \boldsymbol{G}_V(\boldsymbol{x}) = \boldsymbol{S}_V * \boldsymbol{x}, \quad (2)$$

where $*$ denotes the convolution operator. We then calculate the image gradient $\boldsymbol{G}(\boldsymbol{x})$ by,

$$\boldsymbol{G}(\boldsymbol{x}) = \sqrt{\boldsymbol{G}_H(\boldsymbol{x})^2 + \boldsymbol{G}_V(\boldsymbol{x})^2}, \quad (3)$$

where the square and square root are element-wise operations on matrix.

3.2. The Benefit of Fusing Image Gradient

Four different architectures for image gradient fusion. We investigate whether fusing image gradient in the network can facilitate denoising. Specifically, we use the Sobel filter to extract image gradient from clean images. We then fuse the resulting image gradient to different layers of a vanilla denoising network, *i.e.*, DnCNN with a global skip connection, denoted as SKDnCNN, to learn image denoising. We try four different architectures, GradNet-I, GradNet-II, GradNet-III, and GradNet-IV: the first three are based on the location of image gradient fusion; the last one uses the image gradient to weight each pixel in the ℓ_1 loss function. The architectures are shown in Fig. 2.

We conduct **proof-of-concept experiment** using image gradient from clean images. The networks are trained on the NYUv2 [29] or SIDD [1] datasets. For NYUv2, we add $\sigma = 15$ and $\sigma = 50$ additive white Gaussian noise (AWGN) to generate synthetic data. For SIDD, the real noise dataset, we report the results on the validation set. From Fig. 3, the following observations are made.

GradNet-II is the most effective. In all three datasets, GradNet-II achieves the highest PSNR. Comparing with GradNet-I, GradNet-II is higher by +1.13dB, +1.61dB, and +0.64dB on the NYUv2 dataset with $\sigma = 15$ and 50, and on the SIDD dataset, respectively. It verifies our assumption that the fusion of first-order derivative information of an image contributes significantly to the performance.

GradNet-I is the second best. Although lower than GradNet-II, GradNet-I improves by +3.76dB, +4.77dB, +1.35dB on NYUv2 for $\sigma = 15$ and 50 while on SIDD comparing with the highest PSNR by the other four methods. These are also significant improvements and it suggests that concatenating a clean image gradient to the noisy image can boost the denoising capacity of the network, despite being not the most effective approach.

GradNet-III and GradNet-IV are similar to BM3D

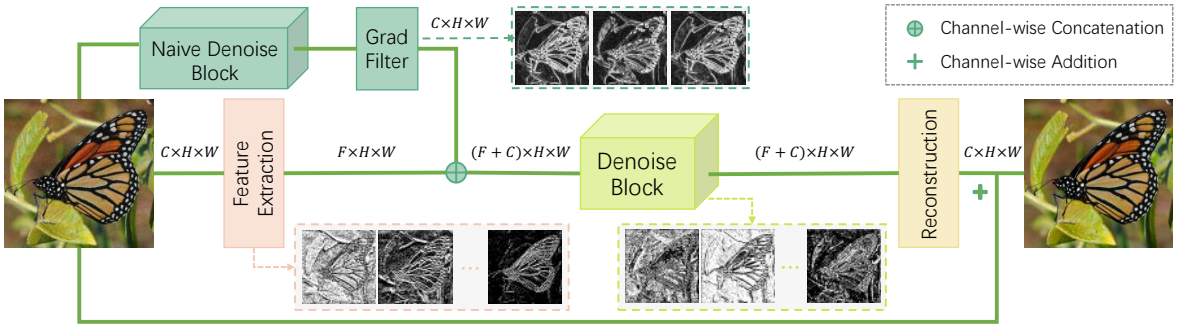


Figure 4: The network structure of GradNet. In the top branch, we obtain a denoised image using a naive denoise block and then calculate its image gradient by the Sobel filter. In the main branch, the feature extraction layer computes F-channel features from the C-channel input image. After concatenation with the image gradient, the $(F + C)$-channel features are passed to the denoise block. The reconstruction layer projects the $(F + C)$-channel feature maps to a C-channel image. Besides, the main branch also has a skip connection. Note that the F-channel feature maps mainly encode edge/texture characteristics, which are reiterated by the explicitly calculated image gradient to be preserved after denoising.

and DnCNN. The difference between GradNet-III and BM3D, as well as SKDnCNN, is no more than $\pm$0.15dB on NYUv2. GradNet-IV is slightly lower than SKDnCNN on the three mentioned datasets. It indicates that fusing image gradient in the last layers essentially offers no difference. The network can not obtain sufficient information while employing the gradient in this fashion.

4. Proposed GradNet

GradNet includes the main denoising branch (backbone), a gradient extraction branch, and a gradient consistency loss (Fig. 4). We describe the three components below.

4.1. Backbone Architecture

The backbone has three blocks, *i.e.*, feature extraction, denoise, and reconstruction, which are similar to typical CNN based denoising methods [5]. Additionally, a long skip connection is used to add the input noisy image to the output of the network. It helps the network to learn the sparse noise and contributes to faster convergence.

Feature extraction block. The feature extraction layer is composed of a convolutional layer followed by ReLU to extract multi-channel features from the input noisy image.

Denoise block. We implement a multiple skip connection residual network (MSKResnet) as the denoise block. Its structure is shown in Fig. 5. MSKResnet cascades several residual modules with a long skip connection to pass the shallower layers features forward. Each of the residual modules contains several residual units with short skip connection (abbreviated as ResUnit), a medium skip connection, and an attention module. These multiple skip connections help to preserve image details for reconstruction.

We further add an Squeeze-and-Excitation block [21] as the attention module after the concatenated feature maps to enforce the network to focus on important features. In this module, we use an average pooling layer to extract the global information for each channel of the feature maps. Then, a shrinkage convolution layer is to learn the dependency between channels. To separate different channel features, we also employ a convolutional layer to upsample the features. Since the output after these operations contains both spatial and channel attention information, we multiply it to the input for refining the feature.

The **reconstruction layer** is a convolutional layer changing multi-channel feature maps to images.

4.2. Image Gradient Fusion

The major contribution of this paper is fusing the image gradient in the network, the motivation of which has been sufficiently demonstrated in the proof-of-concept experiment (Section 3.2). This fusion consists of two parts: where in the network to add the image gradient and how to acquire the clean image gradient.

According to the experiment in Section 3.2, the location of GradNet-II in our architecture boosts the denoising accuracy the most. So **we propose to fuse the image gradient in the shallow layer of the network at the same location with GradNet-II.** However, the clean image gradients are unavailable during testing; some of our attempts show that if we simply train with the clean gradient but test with the noisy gradient, the denoising results will be severely degraded. So, instead, we obtain an approximation of the clean image gradients for training. **We use a denoising method to obtain the approximate clean image first and fusing its gradient in the network**. This is illustrated as the top branch in Fig. 4. The denoising method in the mentioned branch can be different from the method used in the main branch. To distinguish them, we call it naive denoise block here. It could be traditional methods (*e.g.* BM3D), or CNN based methods (*e.g.* DnCNN). A good naive denoised

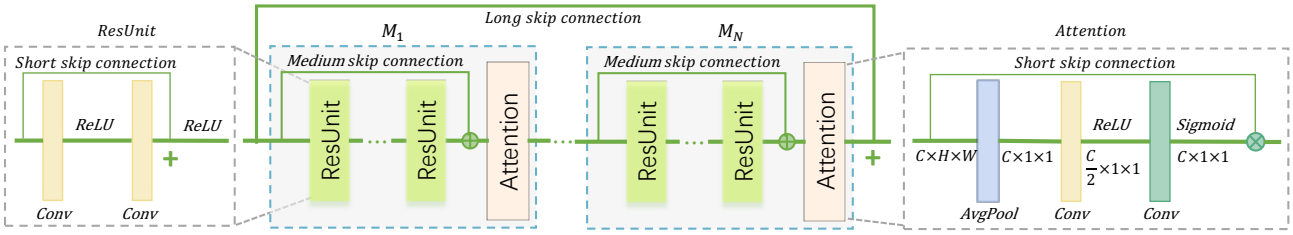


Figure 5: Structure of the multiple skip connection Resnet (MSKResnet). This structure contains N residual modules $M_i, i = 1, ..., N$. Each module has m ResUnits and an attention module [5]. ResUnit is the basic two convolutional block with a short skip connection added before activation ReLU. The attention module consists of an average pooling layer, a shrinkage convolution layer followed by ReLU, and a reconstruction convolution layer with Sigmoid as the activation function. The size of the feature maps is reduced from $C \times H \times W$ to $C \times 1 \times 1$ after the average pooling layer. The shrinkage convolution layer downsamples the channels from C to $\frac{C}{2}$. The reconstruction convolution layer upsamples the features from $\frac{C}{2} \times 1 \times 1$ to $C \times 1 \times 1$ to separate different channel features.

result will definitely improve the performance of GradNet.

Discussions. One might argue that this fusion strategy is trivial, as it looks as merely as a combination of the image gradient with the feature maps in the backbone architecture. However, as indicated in Section 3.2, the performance of fusing with different layers varies significantly. To allow effective fusion, we need to consider the nature of the image gradient: it is in a spirit similar to the output feature maps of the early neural network layers, where the convolutional filters respond specifically to low-level stimuli like edges. Therefore, by fusing the image gradient in the shallow layers, we are concatenating it with its peers that are homogeneous in their physical meaning.

Similarly, one might also doubt the effectiveness of fusing the approximate clean image gradient, which is different from fusing a clean one. Actually, the denoising accuracy with a clean image gradient fused is the upper bound of our capacity. We also add the gradients of noisy images directly to the same position as GradNet-II and get PSNR=36.34, 30.26, and 37.21 for NYUv2 with $\sigma = 15$ and $\sigma = 50$ and SIDD separately. These results are the lower bound of GradNet. However, compared with the results of SKDnCNN reported in Figure 3, these results are still no less than (even a little bit higher than) SKDnCNN without adding image gradient. The results with approximate clean image gradient fused will lie between the lower bound and upper bound, which will definitely achieve improvements. Besides, the better the naive denoise performs, the closer approximate gradient to the clean one, and the higher the denoising results can get. Although we can not achieve as high performance as fusing the clean image gradient, this fusion branch still contributes to reserving edge and texture details while denoising. Our reason is as follows. In denoising tasks, the corrupted input images degrade the features obtained by shallow layers. Adding an approximate clean image gradient explicitly to these features strengthen the edge and texture information exposed to the network, which helps to restore image details.

4.3. Objective Function

Assume the input-output training pairs of our network are $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}_{i=1}^{N}$ which are related to each other through $\boldsymbol{y_i} = \boldsymbol{x_i} + \boldsymbol{n_i}$. $\boldsymbol{x_i}$ is the clean image and $\boldsymbol{y_i}$ is the corresponding noisy image. Let $\hat{\boldsymbol{x_i}}$ to be the naive denoised image and $\widetilde{\boldsymbol{x_i}}$ to be the output of our network. We use the ℓ_1 loss as the objective of the backbone network, GradNet$(\cdot)$,

$$L_p = \frac{1}{N}\sum_{i=1}^{N} ||\text{GradNet}(\boldsymbol{y_i}, G(\hat{\boldsymbol{x}}_{\boldsymbol{i}})) - \boldsymbol{y_i}||_1, \tag{4}$$

where L_p is the pixel loss and $G(\cdot)$ is the gradient filter we mentioned before.

Gradient Consistency loss. Our minor contribution is introducing the gradient consistency loss. We compute the first-order derivatives and employ them as a regularizer, which penalizes the incorrect estimation of the image gradients. Utilizing this regularizer in naive denoising helps to estimate a better approximate image gradient. Furthermore, the mentioned regularization also boosts the denoising accuracy of the main denoise branch via enforcing the semantic attributes in the denoised image, hence forcing it to be as close as possible to the ground-truth. This regularization is denoted in our formulation as L_g and computed as,

$$\begin{aligned} L_g(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}) = &||G_H(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}) - G_H(\boldsymbol{x_i})||_1 \\ &+ ||G_V(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}) - G_V(\boldsymbol{x_i})||_1. \end{aligned} \tag{5}$$

The overall objective function of our network is,

$$L = L_p + \lambda * L_g(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}), \tag{6}$$

where λ is the weight of our gradient consistency loss. The impact of this hyperparameter on the system will be demonstrated in Section 5.2.

Method	CBSD68			Kodak24			McMaster		
σ=	15	25	50	15	25	50	15	25	50
CBM3D [24]	33.50	30.69	27.37	34.28	31.68	28.46	34.06	31.66	28.51
CDnCNN [40] [41]	33.89	31.33	27.97	34.48	32.03	28.85	33.44	31.51	28.61
FFDNet [41]	33.87	31.21	27.96	34.63	32.13	28.98	34.66	32.35	29.18
MSKResnet (Ours)	34.01	31.26	28.06	34.65	32.07	28.99	34.65	32.22	29.12
GradNet (Ours)	**34.07**	**31.39**	**28.12**	**34.85**	**32.35**	**29.23**	**34.81**	**32.45**	**29.39**

Table 1: Comparison with the state-of-the-art approaches on three synthetic datasets with AWGN. Our backbone MSKResnet is competitive comparing with FFDNet. With image gradient fused, GradNet reports further improvements.

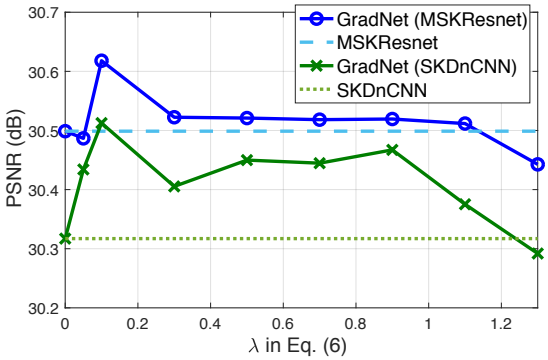


Figure 6: The impact of λ (Eq. 6) on the denoising accuracy (PSNR). We show the results obtained by two GradNets, built on MSKResNet and SKDnCNN, respectively (solid lines). We also provide the baseline results of MSKResNet and SKDnCNN (dashed lines) for reference. From the results, we set $\lambda = 0.1$ in our experiment.

5. Experiment

5.1. Experimental settings

Train/test datasets. This paper evaluates the noisy images synthetically corrupted by spatially invariant AWGN. In training, we use 900 images from DIV2K [2] and 432 images from BSD [28], resulting in 1,332 training images. In each training epoch, we randomly crop four 80×80 patches from each training image and apply data augmentation, including rotations of 90/180/270 degrees, as well as flipping horizontally or vertically. We test our model on the CBSD68 [28], Kodak24 [17], and McMaster [42] datasets.

The real-world noise is more complicated than AWGN, such as being spatially variant (non-Gaussian), signal-dependent, and varying with different camera equipment. We also evaluate our method on three real-world datasets, *i.e.*, DnD [32], SIDD [1] and RNI15 [25]. Since the test sets are diverse and complicated, we combine training images from different datasets as our training set to increase the diversity of noise, edge, and texture details. Specifically, we randomly generate a training set containing 2,000 images of size 512×512 from three real noise datasets, *i.e.* SIDD [1], PolyU [36] and Renoir [4]. They are randomly cropped into 80×80 patches during training.

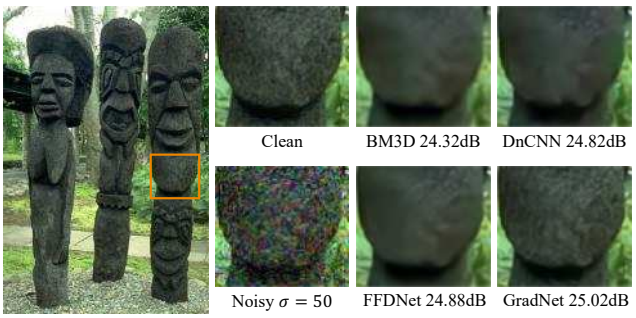
Training details. We choose DnCNN as the naive denoise method and MSKResNet with four residual modules as the main denoise block. Each residual module consists of four ResUnits. The training loss for both pre-trained naive denoise block and the main denoise structure is ℓ_1 loss with gradient loss as regularization. Adam, with default parameters, is employed as the optimizer. The learning rate is initialized as 10^{-3} and decays to 2×10^{-4} after 60 epochs. When the average PSNR of the validation set stops improving for 20 epochs, the learning rate reduces to a fifth of itself. The training process terminates when the learning rate decays to less than 10^{-6}.

5.2. Evaluation

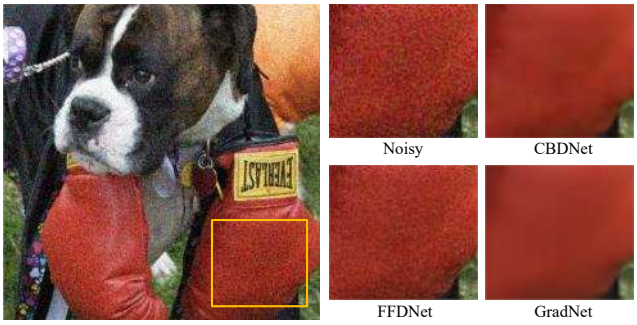
Parameter analysis. We evaluate the impact of hyperparameter λ of the objective funtion on GradNets with SKDnCNN and MSKResnet as the main denoise block separately. The results are shown in Fig. 6. $\lambda = 0$ represents the results obtained without gradient consistency loss. In both backbones, the results are boosted when λ is in [0.1, 1.1] and achieves a peak at 0.1. This is due to the gradient consistency loss is usually one order higher than the main ℓ_1 loss in our experiment. For a large λ, the gradient loss dominates in the objective function. However, we aim to restore the original clean image. A more accurate estimation of its first-order derivative is not necessarily lead to this. We still hope the ℓ_1 loss dominates. On the other hand, a smaller λ may compromise the regularization of gradient consistency loss. So far, $\lambda = 0.1$ is an appropriate parameter we have ever tried, and we apply it in our experiment.

Quantitative results. The comparisons of removing AWGN are shown in Table 1. The backbone MSKResnet reports competitive results on CBSD68 but a bit lower than FFDNet on Kodak24 and McMaster. GradNet outperforms the other three methods consistently on these three datasets. In CBSD68, we improve +0.18dB, +0.06dB, +0.15dB compared with CDnCNN on the three noise levels ($\sigma = 15, \sigma = 25, \sigma = 50$). On Kodak24, GradNet achieves more than +0.22dB higher than FFDNet. We also obtain an increase between 0.1dB-0.21dB on McMaster.

The results of DnD [32] and SIDD [1] are listed in Table 2. MSKResnet achieves competitive performance on SIDD. Based on it, GradNet further improved 0.2dB on both datasets. Especially, we improves 4dB comparing with



(a) An example of textured region denoisng from CBSD68 [28].



(b) An example of smooth region denoising from RNI15 [25].

Figure 7: Visual examples of denoising results on (a) a textured region and (b) a smooth region. In (a), PSNR produced by GradNet is 0.14dB higher than the state-of-art FFDNet. Particularly, GradNet preserves much finer scaled textures on the statue while the other three methods smooth out the details. In (b), the GradNet removes the noise clearly without generating artifacts. Notably the shade in the original image is reserved well at the same time, indicating that GradNet does not simply smooth out the pixels.

Method	DnD		SIDD	
	PSNR	SSIM	PSNR	SSIM
DnCNN [40]	32.43	0.7900	23.66	0.583
TNRD [12]	33.65	0.8306	24.73	0.643
FFDNet [41]	34.40	0.8474	–	–
BM3D [14]	34.51	0.8507	25.65	0.685
NI	35.11	0.8778	–	–
NC [26]	35.43	0.8841	–	–
KSVD [3]	36.49	0.8978	26.88	0.842
TWSC [37]	37.96	0.9416	–	–
CBDNet [19]	38.06	0.9421	33.28	0.868
CIMM [6]	39.20	0.9524	–	–
MLDN	39.23	0.9516	–	–
RIDNet [5]	39.26	0.9528	–	–
PRIDNet [43]	39.42	0.9528	–	–
DRDN [35]	39.43	0.9531	–	–
DeepProxies BM3D	–	–	34.34	0.911
MSKResnet (ours)	39.25	0.9534	38.10	0.946
GradNet (ours)	**39.44**	**0.9543**	**38.34**	**0.946**

Table 2: Comparisons on the real noise datasets.

the published state-of-art DeepProxies BM3D on SIDD. For DnD, although GradNet only has a slight advantage over DRDN [35] measured by PSNR, we obtain higher accuracy measured by SSIM. This measurement for structure similarity illustrates our superiority in preserving the detailed structures. On the other hand, the parameter numbers of the GradNet we use is 1.6×10^6, which is much smaller than the other competitive methods, like DRDN (5.9×10^6 parameters, about 3.3 times of ours) and PRIDNet (7.4×10^6 parameters, about 4.1 times of ours). We achieve competitive results in a cost-effective way.

Backbone	Image Grad.	Grad. Loss	PSNR
SKDnCNN			30.0950
		✓	30.1545
	✓		30.2084
	✓	✓	30.3862
MSKResnet			29.8449
		✓	29.9742
	✓		30.4833
	✓	✓	30.5885

Table 3: Evaluation of the benefit of image gradient fusion and gradient loss. ✓means that the gradient loss or the image gradient fusion is integrated into the network.

Naive Denoise Block	Main Denoise Block	PSNR
SKDnCNN	–	30.0950
	MSKResent	30.7096
MSKResnet	–	29.8449
	MSKResent	30.5919

Table 4: Evaluation of the effect of naive denoise block.

Qualitative performance. We visually compare our method with others on a synthetic noise example and a real world noise example in Fig. 7. In the examples, our method preserves fine-scale textured and smooth regions better than other competitive methods.

5.3. Ablation study

In this part, we study the effect of image gradient and gradient loss. We trained two basic architectures, *i.e.* SKDnCNN and MSKResnet, on the synthetic training set mentioned earlier with $\sigma = 50$ AWGN. Each of the models is trained 30 epochs with patch size 80×80. The learning rate is initialized as 10^{-3} and decayed to 2×10^{-4} after 20 epochs. The results are illustrated in Table 3. Training with gradient loss as regularization ($\lambda = 0.1$) improves 0.06dB on SKDnCNN and 0.13dB on MSKResnet. In addition, adding an image gradient module to these two structures promotes 0.11dB and 0.64dB separately. Overall, employing both gradient loss and gradient module yields improvements of 0.29dB and 0.74dB.

We also study the influence of naive denoise blocks. We compare the results of different naive denoise blocks (SKDnCNN and MSKResnet) and the same main denoise block (MSKResnet). Both models are also trained for 30 epochs with the same training strategy mentioned above. The results are reported in Table 4. With a higher naive denoise result by SKDnCNN (0.25dB higher than MSKResnet), the

Clean Textured Regions

Figure 8: Illustration of selecting smooth/textured regions(Section 5.4). For the image on the left, we use white lines to delineate its textured regions on the right. Its smooth regions are masked with a transparent gray channel.

corresponding GradNet gets a higher final denoise accuracy (0.12dB higher than the other). The results are consistent with our inference in a proof-of-concept that the higher the naive denoise block gets, the better the final denoise result.

5.4. Working Mechanism Analysis

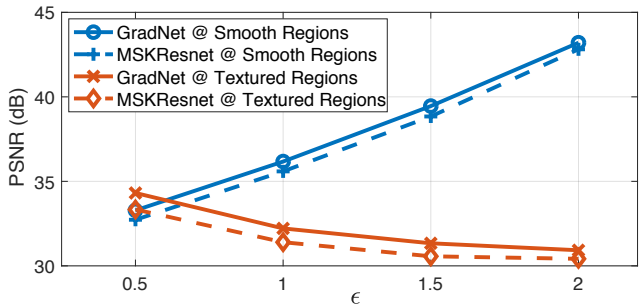
We further study the models obtained in Section 5.3 on MSKResnet. For each image in the test set of NYUv2, we use a threshold to divide it into smooth regions and textured regions. Then we calculate the PSNR in these two regions to examine the variation of the performance of GradNet.

In our experiment, the values of the gradient in each pixel of normalized images (pixel values $\in [0, 1]$) are between [0, 4.5]. We set a threshold ϵ manually in this range for the division. That is, when a pixel value of a clean image gradient is above ϵ, we merge this pixel with the other eight pixels around it to form a textured region. The textured regions acquired in this way consists of several disconnected regions, while the remaining pixels are grouped into smooth regions, as shown in Fig. 8. Different thresholds, $\epsilon = 0.5$, 1, 1.5, and 2, are tried in our experiment to obtain different divisions of regions. The higher ϵ is, the less smooth details are contained in textured regions *i.e.* the proportion of edges and textures increases. From the results illustrated in Fig. 9, the following observations can be implied.

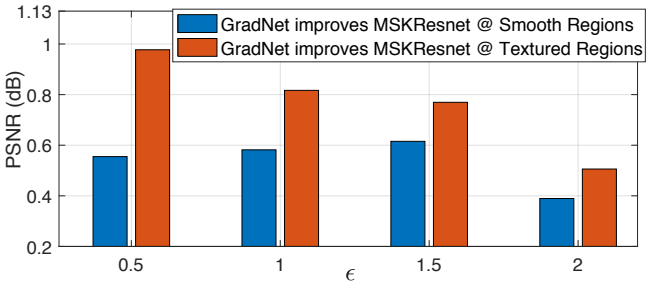
Textured regions are more challenging than smooth regions. In Fig. 9a, with an increase of ϵ, the trend of PSNR in smooth regions is upward, while it goes down in textured regions. This illustrates that the denoising accuracy of both methods in textured regions is significantly worse than smoother regions, illustrating that restoring texture regions is a more challenging task.

GradNet outperforms vanilla MSKResnet in both textured and smooth regions. Under different region divisions, the performance of GradNet is consistently above the performance of vanilla MSKResnet. This represents the superiority of GraNet in both dealing with smooth and challenging regions.

GradNet improves more remarkably in textured regions than smooth regions. The improvement of GradNet



(a) Comparison of GradNet and its baseline MSKResnet at smooth and textured regions.



(b) Improvement of GradNet over its baseline MSKResnet at smooth and textured regions

Figure 9: GradNet in smooth vs. textured regions. ϵ controls the magnitude of the image gradient in the textured regions. A larger ϵ means that textured regions have a more significant image gradient. In the first graph, we compare GradNet and its baseline, MSKResnet, in their denoising accuracy in the two types of regions. In the bottom bar plot, we show the PSNR improvement of GradNet over MSKResnet in textured (red) and smooth (blue) regions. GradNet brings about a more significant improvement in textured regions than in smooth regions.

in smooth and textured regions are plotted in Fig. 9b. The bars which represent the improvement in texture regions are consistently higher than the improvement in smooth regions. This confirms that adding an image gradient contributes to the restoration of edges and textures.

6. Conclusions

This paper introduces GradNet that preserves the high-frequency textures and edges when denoising. Our main contribution is to fuse the image gradient in the shallow layer features. The four architectures in concept proof validate the effectiveness of this fusion. We also propose a gradient consistency regularizer as a supplementary of GradNet. This regularizer minimizes the difference between the gradient of the denoised image and the corresponding clean image gradient. Experiment on 3 synthetic datasets and 3 real datasets demonstrates the competitive denoising accuracy of GradNet. Importantly, we verify that GradNet is more advantageous in removing noise from textured regions than smooth regions, hence validating our hypothesis.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, June 2018. 2, 3, 6

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

[3] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006. 7

[4] Josue Anaya and Adrian Barbu. Renoir - a dataset for real low-light noise image reduction. *arXiv preprint arXiv:1409.8230*, 2014. 6

[5] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. *CoRR*, abs/1904.07396, 2019. 4, 5, 7

[6] Saeed Anwar, Cong Phuoc Huynh, and Fatih Porikli. Chaining identity mapping modules for image denoising. *CoRR*, abs/1712.02933, 2017. 7

[7] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. In *CVPR*, June 2019. 2

[8] Benoit Brummer and Christophe De Vleeschouwer. Natural image noise dataset. abs/1906.00270v1, 2019. 2

[9] J. Canny. A computational approach to edge detection. *TPAMI*, PAMI-8(6):679–698, Nov 1986. 2

[10] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *CoRR*, abs/1805.01934, 2018. 2

[11] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, June 2018. 1, 2

[12] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *TPAMI*, 2017. 7

[13] Supakorn Chupraphawan and Chotirat Ann Ratanamahatana. Deep convolutional neural network with edge feature for image denoising. In Pongsarun Boonyopakorn, Phayung Meesad, Sunantha Sodsee, and Herwig Unger, editors, *Recent Advances in Information and Communication Technology 2019*, pages 169–179, Cham, 2020. Springer International Publishing. 1, 2

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 16(8):2080–2095, Aug 2007. 7

[15] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *TIP*, 22(4):1620–1630, 2013. 1

[16] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 15(12):3736–3745, 2006. 1

[17] Robert Franzén. Kodak lossless true color image suite: Photocd pcd0992. 2002. 6

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

[19] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *CVPR*, 2019. 1, 2, 7

[20] F. Hosotani, Y. Inuzuka, M. Hasegawa, S. Hirobayashi, and T. Misawa. Image denoising with edge-preserving and segmentation based on mask nha. *TIP*, 24(12):6025–6033, Dec 2015. 2

[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2017. 4

[22] Paras Jain and Vipin Tyagi. A survey of edge-preserving image denoising methods. *Information Systems Frontiers*, 18(1):159–170, Feb 2016. 2

[23] Birgit Komander, Dirk A. Lorenz, and Lena Vestweber. Denoising of image gradients and total generalized variation denoising. *Journal of Mathematical Imaging and Vision*, 61(1):21–39, May 2018. 2

[24] Vladimir Katkovnik Kostadin Dabov, Alessandro Foi and Karen Egiazarian. Image denoising by sparse 3d transform-domain collaborative ltering. *TIP*, 2007. 6

[25] M. Lebrun, M. Colom, and J. M. Morel. The noise clinic: A universal blind denoising algorithm. In *ICIP*, pages 2674–2678, Oct 2014. 6, 7

[26] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The Noise Clinic: a Blind Image Denoising Algorithm. *Image Processing On Line*, 5:1–54, 2015. 7

[27] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*, 2016. 1

[28] 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 *ICCV*, volume 2, pages 416–423, July 2001. 6, 7

[29] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3

[30] Muhammad Nawaz. Variational regularization for multichannel image denoising. *Pakistan Journal of Engineering and Technology*, 2(1):51–58, Aug. 2019. 2

[31] Ram Krishna Pandey, Harpreet Singh, and A. G. Ramakrishnan. Improvement of image denoising algorithms by preserving the edges. In Mario Vento and Gennaro Percannella, editors, *Computer Analysis of Images and Patterns*, pages 496–506, Cham, 2019. Springer International Publishing. 1, 2

[32] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR Workshops*, 2017. 2, 6

[33] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. 2018. NeurIIPS. 1

[34] Irwin Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 02 2014. 3

[35] Yuda Song, Yunfang Zhu, and Xin Du. Dynamic residual dense network for image denoising. *Sensors*, 19(17):3809, Sep 2019. 7

[36] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *CoRR*, abs/1804.02603, 2018. 2, 6

[37] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. *CoRR*, abs/1807.04364, 2018. 7

[38] Y.Shen, Q.Liu, S.Lou, and Y.Hou. Wavelet-based total variation and nonlocal similarity model for image denoising. *IEEE Signal Processing Letters*, 24(6):877–881, June 2017. 2

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

[40] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 26(7):3142–3155, 2017. 1, 6, 7

[41] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *TIP*, 2018. 1, 2, 6, 7

[42] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging - J ELECTRON IMAGING*, 20, 01 2011. 6

[43] Yiyun Zhao, Zhuqing Jiang, Aidong Men, and Guodong Ju. Pyramid real image denoising network, 2019. 7

[44] W. Zuo, L. Zhang, C. Song, and D. Zhang. Texture enhanced image denoising via gradient histogram preservation. In *CVPR*, pages 1203–1210, June 2013. 1, 2