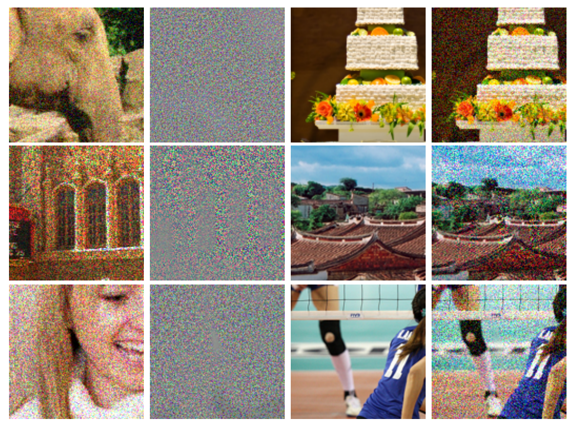
Noise2Grad: Extract Image Noise to Denoise

Huangxing Lin[1] , **Yihong Zhuang**[1] , **Yue Huang**[1] , **Xinghao Ding**[1*] ,
Xiaoqing Liu[2] and **Yizhou Yu**[3]
[1]Xiamen University, China
[2]Deepwise AI Lab
[3]The University of Hong Kong
{hxlin, zhuangyihong}@stu.xmu.edu.cn, {yhuang2010, dxh}@xmu.edu.cn,
Liuxiaoqing@deepwise.com, yizhouy@acm.org

Abstract

In many image denoising tasks, the difficulty of collecting noisy/clean image pairs limits the application of supervised CNNs. We consider such a case in which paired data and noise statistics are not accessible, but unpaired noisy and clean images are easy to collect. To form the necessary supervision, our strategy is to extract the noise from the noisy image to synthesize new data. To ease the interference of the image background, we use a noise removal module to aid noise extraction. The noise removal module first roughly removes noise from the noisy image, which is equivalent to excluding much background information. A noise approximation module can therefore easily extract a new noise map from the removed noise to match the gradient of the noisy input. This noise map is added to a random clean image to synthesize a new data pair, which is then fed back to the noise removal module to correct the noise removal process. These two modules cooperate to extract noise finely. After convergence, the noise removal module can remove noise without damaging other background details, so we use it as our final denoising network. Experiments show that the denoising performance of the proposed method is competitive with other supervised CNNs.



(a) (b) (c) (d)

Figure 1: (a): Noisy images. (b): Noise maps extracted from (a) by our method. (c): Random clean images. (d): Noisy images obtained by multiplying (b) with a random binary (-1 or 1) mask and then superimposed on (c). The noise in (d) is similar to that in (a). From top to bottom the noise are Gaussian, Speckle and Poisson.

1 Introduction

Removing noise from an image is a preprocessing step in many imaging projects to facilitate visualization and downstream tasks such as image segmentation and detection. A noisy image x can be represented as

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

where n denotes the measurement noise, y is the clean image to be restored. This inverse problem is challenging because the statistics of n are usually unknown and complex.

1.1 Related Work

In the past decades, image denoising has been an active research topic in image processing. Existing image denoising methods can be roughly divided into model-based methods and learning-based methods.

Model-based methods. Most classical image denoising algorithms exploit hand-crafted priors [Xu *et al.*, 2018], [Buades *et al.*, 2005], [Meng and De La Torre, 2013], [Zhao *et al.*, 2014] to guide the denoising process. The non-local self-similarity (NSS) prior [Hou *et al.*, 2020] has demonstrated its powerful ability to aid noise removal. The NSS prior reveals the fact that a natural image contains many similar but non-local patches. Some well-known NSS-based methods include BM3D [Dabov *et al.*, 2007] and WNNM [Gu *et al.*, 2014]. Other prominent techniques, such as wavelet coring [Simoncelli and Adelson, 1996], total variation [Selesnick, 2017] and low-rank assumptions [Zhu *et al.*, 2016] have also been utilized to simplify the denoising problem. Though simple and effective, these model-based methods tend to produce over-smoothed results and cannot handle noise that does not meet their priors.

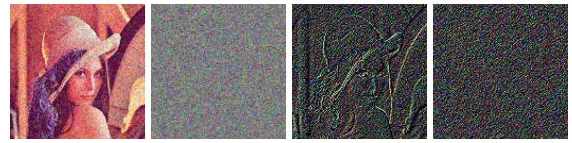
[*]Contact Author

	Gaussian noise ($\sigma = 25$)	Speckle noise ($v = 0.1$)	Poisson noise ($\lambda = 30$)
Original	31.81	31.49	31.09
Fake	31.65	29.90	30.70

Table 1: PSNR results (dB) on BSD300 test set. σ, v and λ represent the noise level.

Learning with paired data. Recently, deep learning has achieved unprecedented success in image denoising. CNNs trained with plenty of noisy/clean image pairs have become the dominant method. One of the seminal networks for image denoising is DnCNN [Zhang *et al.*, 2017], which adopts a residual architecture to ease learning. Beyond DnCNN, many versatile network architectures have been developed to achieve better denoising results, including FFDNet [Zhang *et al.*, 2018], DANet [Yue *et al.*, 2020], NLRN [Liu *et al.*, 2018], VDN [Yue *et al.*, 2019] and RIDNet [Anwar and Barnes, 2019]. These supervised CNNs consistently show impressive performance on some predefined datasets (*e.g.* synthetic datasets). However, in many imaging systems (*e.g.* medical imaging, SAR imaging), paired noisy/clean images are difficult to collect, which limits the application of these supervised techniques. To mitigate this issue, Lehtinen *et al.* [Lehtinen *et al.*, 2018] demonstrate that the denoising CNN can be trained with pairs of independent noisy measurements of the same scene. This elegant training strategy is called Noise2Noise (N2N), and it can achieve denoising performance on par with general supervised learning methods. Nevertheless, it is not always feasible to sample two independent noises for the same scene.

Learning without paired data. To relax the requirement for training data, training CNN denoisers without pre-collected paired data has become a hot topic [Liu *et al.*, 2019]. Inspired by the Noise2Noise method, Noise2Void [Krull *et al.*, 2019] and Self2Self [Quan *et al.*, 2020] further demonstrate that denoising CNNs can be trained with individual noisy images via the blind spot strategy. These methods are self-supervised because they do not rely on external data to provide supervision. Considering its practical value, the blind spot strategy is further improved in [Laine *et al.*, 2019], [Batson and Royer, 2019], [Wu *et al.*, 2020] to achieve better denoising results. Unfortunately, the effectiveness of these self-supervised methods stems from some pre-defined statistical assumptions, for example, the noise n is zero mean and pixel-independent. This means that these methods cannot cope with noise that violates their assumptions, such as spatially correlated noise. Another elegant strategy is to use unpaired noisy and clean images to learn the transformation between the noise domain and the clean domain. To this end, methods to this category usually integrate noise modeling and removal into the same deep learning framework. For instance, generative adversarial network (GAN) [Chen *et al.*, 2018], [Kaneko and Harada, 2020], [Yan *et al.*, 2020] is widely used to synthesize noise samples corresponding to accessible clean images. These generated data then provide supervision for denoising. However, GAN-based methods are easy to suffer from mode collapse, so that the generated noise is usually unrealistic or lacks diversity. These unrealistic and monotonous noises can lead to poor denoising.



(a) (b) (c) (d)

Figure 2: (a): An image with Gaussian noise ($\sigma = 25$). (b): A noise map obtained by subtracting ground-truth from (a). (c): The image gradient of (a), object edges from (a) are retained in (c). (d): The image gradient of (b).

1.2 Motivations

Considering that the collection of unpaired data is easy in most applications and unpaired images contain more information than individual noisy images, the topic of this paper is unpaired denoising. To form supervision, an intuitive idea is to extract noise n from the noisy image x and add it to other clean images to obtain pairs of data. To verify the feasibility of this idea, we first conduct some denoising experiments. We synthesize three noise datasets (original) by software (*i.e.* Gaussian, Speckle and Poisson), and the ground-truth is available. Then, we subtract the ground-truth from these noisy images to obtain noise components, which are superimposed on random clean images to construct new noise datasets (fake). We train denoising U-Nets on these original and fake datasets respectively, and the results are reported in Table 1. As can be seen, the network trained on the "fake" dataset achieves comparable denoising performance with the network trained on the "original" dataset. Even for signal-dependent noise (*i.e.* Speckle and Poisson), this strategy is still feasible. These experiments show the effectiveness of our noise superposition strategy. Based on these observations, the focus of this paper is shifted to how to extract noise from noisy images when ground-truth is not available.

On the other hand, we notice that the gradient of the noisy image is dominated by noise (see Figure 2), which inspires us that the gradient of the noisy image can be used to guide the noise extraction while avoiding the interference of other background content. However, the noisy image gradient contains object edges in addition to noise, which may contaminate the extracted noise. How to counteract the negative effect of image gradient is a thorny problem.

1.3 Our Contributions

Based on the above analysis, we develop a new training strategy *Noise2Grad* to train a noise extraction network. To reduce the interference of the background content of the noisy image, we divide the noise extraction into two subtasks: noise removal and noise approximation. First, the noise removal module roughly removes the noise from the noisy input, and the removed noise is input to a noise approximation module. Since most of the background details have been eliminated by

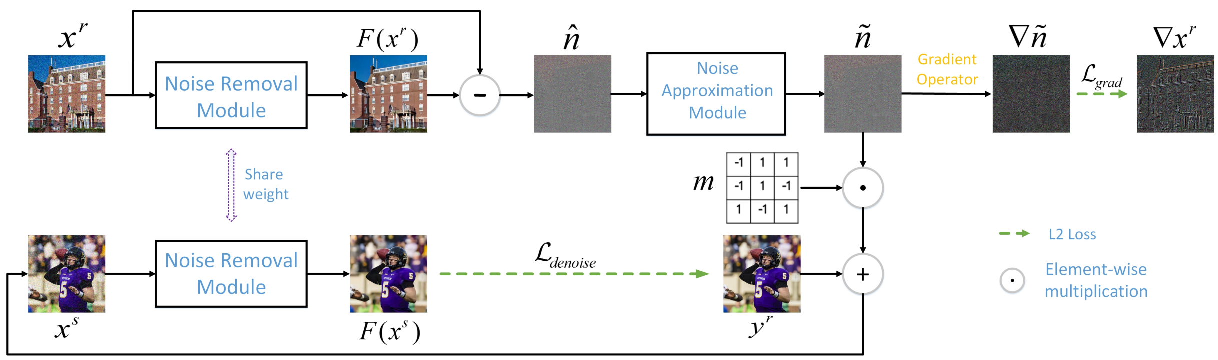


Figure 3: Illustration of Noise2Grad training scheme for noise extraction and denoising. The noise extraction network consists of a noise removal module and a noise approximation module. The dimension of the random binary (-1 or 1) mask m is the same as that of $\tilde{n}$. For better visualization, only nine elements in m are shown.

the noise removal module, the noise approximation module can easily produce a realistic noise map to match the gradient of the noisy image. The noise map is multiplied by a random binary (-1 or 1) matrix to destroy the residual structural details, and then superimposed on a random clean image to synthesize a new noisy / clean image pair. These synthesized data are fed back to the noise removal module to guide better noise removal, which indirectly leads to subsequent finer noise extraction. After convergence, the two modules cooperate to extract noise components from the noisy input. Since the noise removal module is trained with data synthesized by our method, we directly use it as the final denoising network instead of retraining one. Through experiments on several denoising tasks, we demonstrate that the denoising performance of Noise2Grad is close to that of CNNs trained with pre-collected paired data and significantly outperforms other self-supervised and unpaired denoising methods.

2 Methodology

Given some unpaired noisy images $\mathcal{D}^{noise} = \{x_i^r\}_{i=1}^N$ and clean images $\mathcal{D}^{clean} = \{y_j^r\}_{j=1}^M$, the major objective of this paper is extract noise from noisy images and use it to synthesize new noisy data. Hereafter, we use the superscript r to represent real data and s to represent the data synthesized by our method. Our noise extraction network is shown in Figure 3. Since noise extraction is constrained by the image gradient, this training scheme is called Noise2Grad (N2G). In effect, the gradient of the noisy image does not promise high-quality noise extraction. For this problem, the noise extraction network is divided into two parts, a noise removal module and a noise approximation module. These two modules use each other's output to correct their own learning, and cooperate to achieve fine noise extraction.

2.1 Noise Extraction

Image gradient reflects the high-frequency content but excludes the low-frequency part of the image. Generally, the gradient of a noisy image is mainly composed of the gradient of noise. This inspires us that the gradient of the noisy image can be a hint for noise extraction. We calculate the image gradient by combining the horizontal and vertical deviations of adjacent pixels,

$$\begin{aligned}\nabla x &= 0.5 \cdot (\nabla_h x + \nabla_v x) \\ &= 0.5 \cdot \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} * x + \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} * x \right),\end{aligned} \tag{2}$$

where $*$ indicates the convolution operation, ∇ represents the image gradient. Then, the noise extraction network aims to minimize the following gradient similarity loss,

$$\mathcal{L}_{grad} = \sum_i \|\nabla\tilde{n}_i - \nabla x_i^r\|_2^2, \tag{3}$$

where $L2$ loss is adopted. $\tilde{n}$ is the output of noise approximation module, it can be expressed as,

$$\begin{aligned}\tilde{n} &= G(\hat{n}) \\ &= G(x^r - F(x^r)) \\ &= n + \tilde{\varepsilon},\end{aligned} \tag{4}$$

where x^r is the noisy input, n is the noise component of x^r that we want, $\tilde{\varepsilon}$ represents some structural details from x^r, $F(\cdot)$ and $G(\cdot)$ represent noise removal module and noise approaximation module, respectively.

Following Eq.(3), the gradient of $\tilde{n}$ is always similar to that of the noise component n. However, some structural details may remain in $\tilde{n}$ (*i.e.* $\|\tilde{\varepsilon}\| \neq 0$), because the label ∇x^r contains object edges in addition to noise. Our goal is to obtain the noise component n, so we have to remove the residual structural detail $\tilde{\varepsilon}$ in $\tilde{n}$. Fortunately, this can be achieved by reducing the structural information contained in the input of the noise approximation module. If the input of the noise approximation module is only noise without other structural information of x^r (*i.e.* $\hat{n} = n$), then its output $\tilde{n}$ should be mainly composed of noise, and other structural details are unlikely to be retained. To do this, we apply a simple feedback technique to turn $\hat{n}$ into a "*clean*" noise. We add the noise map $\tilde{n}$ to a random clean image y^r to synthesis a new noisy image x^s,

$$x^s = y^r + \tilde{n}. \tag{5}$$

x^s and x^r have similar noise, but their image content is different. More importantly, x^s has a clean label y^r, so they can be paired to guide denoising. x^s is fed back to the noise removal module to obtain a denoised result $F(x^s)$. The corresponding noise removal objective function is

$$\mathcal{L}_{denoise} = \sum_i \|F(x_i^s) - y_i^r\|_2^2 . \tag{6}$$

Combining Eqs.(3) and (6), the overall loss function is,

$$\mathcal{L} = \mathcal{L}_{grad} + \mathcal{L}_{denoise}. \tag{7}$$

In this setting, the noise removal module learns to restore image details while removing noise. This means that the $\hat{n}$ removed by the noise removal module is mainly composed of noise *i.e.*

$$\begin{aligned} \hat{n} &= x^r - F(x^r) \\ &= n + \hat{\varepsilon}, \end{aligned} \tag{8}$$

where $\hat{\varepsilon}$ denotes some structural details of x^r. Since much image background information is excluded, the noise approximation module is easier to produce realistic noise map. With the noise removal module doing better in restoring image details, we want $\|\hat{\varepsilon}\|$ to converge to 0.

2.2 Detail Destruction

At each training step t, a new noisy image x^s can be obtained by noise superposition Eq.(5). Since $\tilde{n}$ may contain some structural details $\tilde{\varepsilon}$ from x^r, the synthesized noisy image x^s may not be realistic. The unrealistic x^s may impair the ability of the noise removal module to restore image details and remove noise. To solve this problem, $\tilde{n}$ is multiplied by a random binary mask m before being superimposed on the clean image y^r. Eq.(5) can be reformulated as

$$x^s = y^r + \tilde{n} \odot m. \tag{9}$$

where $\odot$ represents element-wise multiplication, m has the same dimension as $\tilde{n}$. Each element in m is set to 1 with a probability of 0.5, otherwise is -1. The random mask m can destroy the residual structural information in the $\tilde{n}$ to obtain a more realistic x^s (see Figure 4, (l)-(n)). The realistic x^s is beneficial for the noise removal module to learn noise removal and image detail restoration. In addition, multiplying with m can be seen as a way of data augmentation. Since m is random, various noises can be obtained.

2.3 Update Delay

As the noise removal module becomes better at restoring image details while removing noise, the structural information $\hat{\varepsilon}$ remaining in $\hat{n}$ gradually decays. Since the noise approximation module takes $\hat{n}$ as the input, the weakening of $\hat{\varepsilon}$ will also lead to the attenuation of $\tilde{\varepsilon}$ in $\tilde{n}$ (see Figure 4, (b)-(d) and (e)-(g)). However, the gradient similarity loss $\mathcal{L}_{grad}$ Eq.(3) hinders this convergence process due to the edge information in the the noisy image gradient (*i.e.* ∇x^r). To alleviate the negative impact of $\mathcal{L}_{grad}$, we gradually reduce the frequency of calculating $\mathcal{L}_{grad}$, that is, the time interval τ for calculating $\mathcal{L}_{grad}$ becomes longer. τ can be further expressed as

$$\tau = \lfloor \frac{t}{500} \rfloor + 1, \tag{10}$$

where t denotes the training step, $\lfloor \cdot \rfloor$ is the floor function.

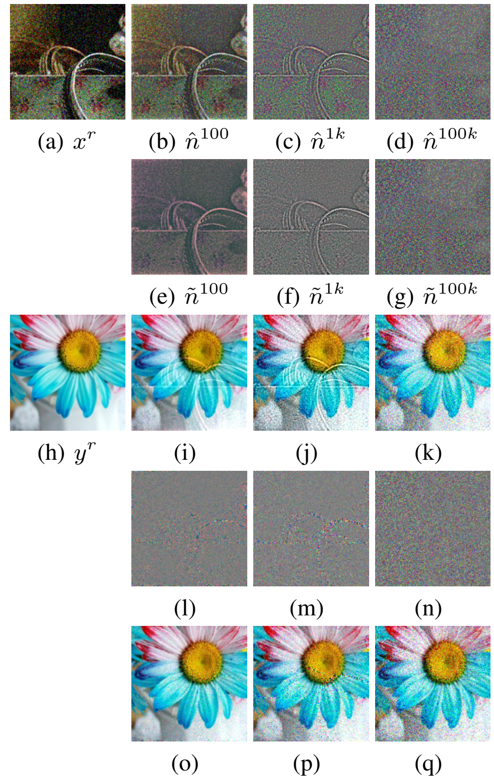


Figure 4: Visual examples produced by our noise extraction network during training. (a): A noisy image. (b)-(d): Noise maps produced by the noise removal module. The superscript represents the number of training steps. (e)-(g): Noise maps corresponding to (b)-(d) produced by the noise approximation module. (h): A clean image. (i)-(k): New noisy images obtained by superimposing (e)-(g) to (h) respectively. (l)-(n): New noise maps obtained by multiplying (e)-(g) with a random binary mask m respectively. (o)-(q): New noisy images obtained by superimposing (l)-(n) to (h) respectively.

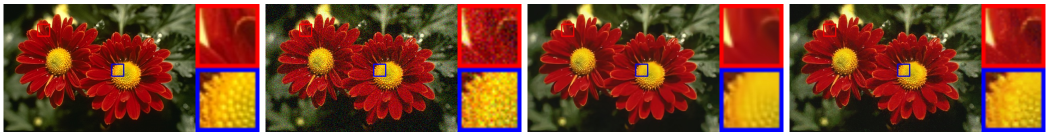
By gradually delaying the calculation of $\mathcal{L}_{grad}$, the noise removal module can focus on preserving image details while removing noise. After convergence, the $\hat{n}$ removed by the noise removal module contains little structural details. Due to the lack of sufficient structural information, the noise approximation module can only output a realistic noise map $\tilde{n} = G(\hat{n}) \approx n$.

Note that, we use the noise removal module as our final denoising network.

2.4 Architecture and Training Details

The implementation of N2G is based on CNNs. For simplicity, we adopt a simple U-Net [Ronneberger *et al.*, 2015] as the noise removal module, while the noise approximation module is only composed of one 1×1 convolution layer. We use PyTorch and Adam with a batch size of 1 to train the network. The training images are randomly cropped into 128×128 patches before being input to the network. The learning rate if fixed to 0.0002 for the first $2,500,000$ iterations and linearly decays to 0 for the next $2,500,000$ iterations.[1]

[1]We will release our code and datasets soon.



(a) Clean | SSIM, PSNR (b) Input | 0.504, 24.59 (c) BM3D | 0.904, 31.71 (d) NLH | 0.846, 30.84

(e) N2V | 0.911, 32.09 (f) LIR | 0.756, 18.29 (g) U-Net | **0.932, 34.06** (h) Our | 0.928, 33.79

Figure 5: Example results for Gaussian denoising, $\sigma = 25$.

	Test noise level	BM3D	NLH	N2V	S2S	LIR	N2N	U-Net	N2G
Gaussian	$\sigma = 25$	30.90	30.31	30.56	29.63	26.91	31.62	**31.81**	31.71
	$\sigma \in (0, 50]$	31.69	31.77	31.88	27.76	26.38	33.27	**33.44**	33.36
Speckle	$v = 0.1$	26.64	25.18	28.40	27.60	25.66	31.13	**31.49**	29.84
	$v \in (0, 0.2]$	26.70	26.10	28.77	27.53	25.44	31.39	**31.86**	30.16
Poisson	$\lambda = 30$	27.70	28.49	29.78	29.23	26.15	30.91	**31.09**	30.64
	$\lambda \in [5, 50]$	27.23	26.12	28.94	28.01	25.62	30.25	**30.44**	29.64

Table 2: PSNR results (dB) from BSD300 dataset for Gaussian, Speckle and Poisson noise. **Bold**: best. Red: second. Blue: third.

3 Experiments

We evaluate the effectiveness of Noise2Grad on several image denoising tasks in this section.

3.1 Synthetic Noises

Our Noise2Grad requires some unpaired noisy and clean images to train the network. We use the 4744 natural images in [Ma *et al.*, 2016] to synthesize noisy images (*i.e.* D^{noise}) with software. In addition, 5000 clean images collected from the Internet are adopted as the clean set D^{clean}. We compare N2G with several state-of-the-art denoising methods, including model-based methods BM3D [Dabov *et al.*, 2007] and NLH [Hou *et al.*, 2020], self-learning methods Noise2Void(N2V) [Krull *et al.*, 2019] and Self2Self(S2S) [Quan *et al.*, 2020], an unpaired learning method LIR [Du *et al.*, 2020], other deep learning methods include Noise2Noise(N2N) [Lehtinen *et al.*, 2018] and a common fully-supervised U-Net. For the sake of fairness, N2N, U-Net and our N2G adopt the same network architecture to perform denoising. BSD300 [Martin *et al.*, 2001] is the test set of the following experiments.

Gaussian noise. We first conduct the comparative experiments on additive gaussian noise. We add zero-mean gaussian noise with a random standard deviation $\sigma \in (0, 50]$ to each training example. For the test sets, noisy images are synthesized in two ways: a fixed noise level $\sigma = 25$ and a variable $\sigma \in (0, 50]$. PSNR comparisons are reported in Table 2. Subjective denoising results are shown in Figure 5.

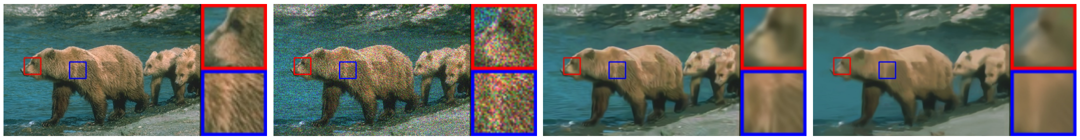
Speckle noise. Multiplicative speckle noise, often observed in medical ultrasonic images and radar images. It is harder to remove than Gaussian noise because it is signal dependent. This noise can be modeled by random value multiplications with pixel values of the latent singal y and can be expressed as $x = y + y \cdot n$. In this model, n is an uniform noise with a mean of 0 and a variance of v. We vary $v \in (0, 0.2]$ to synthesize training images. Quantitative and visual comparisons are shown in Table 2 and Figure 6, respectively.

Poisson noise. We then consider poisson noise, which can be used to model photon noise in imaging sensors. The expected magnitude of poisson noise is dependent on the pixel brightness. Following the setting in [Laine *et al.*, 2019], we randomize the noise level $\lambda \in [5, 50]$ separately for each training example. We show the comparisons in Table 2.

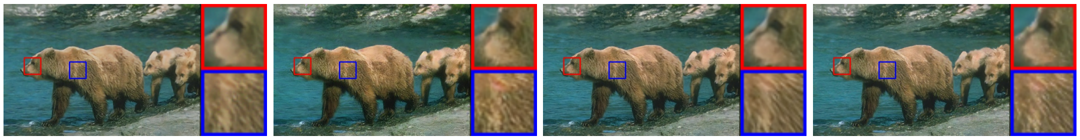
Discussion. From the above experiments, we see that the denoising performance of N2G is close to supervised methods (U-Net and N2N), and significantly outperforms other denoising methods (*e.g.* BM3D and N2V). The effectiveness of N2G does not rely on paired data or prior knowledge about the noise. The only prerequisite is some unpaired noisy and clean images, which is easy to achieve in most practical applications. For various types of noise, N2G consistently exhibits satisfactory denoising performance. The denoised images are clean and sharp. Besides, the denoising results of N2G are close to the results in Table 1 (Fake). This indicates that the noise extracted by N2G is similar to the real noise component of the noisy image. These experiments show that N2G is a promising solution for various denoising tasks.

3.2 Ablation Study

Detail destruction and update delay and are two key components of the N2G training scheme. We then explore the impact of these two technologies on the performance of N2G.



(a) Clean | SSIM, PSNR (b) Input | 0.453, 21.27 (c) BM3D | 0.698, 28.28 (d) NLH | 0.582, 26.41



(e) N2V | 0.832, 30.53 (f) LIR | 0.794, 28.38 (g) U-Net | **0.886**, **31.94** (h) Our | 0.860, 31.00

Figure 6: Example results for Speckle denoising, $v = 0.1$.

		N2G^D	N2G^U	N2G
Gaussian	SSIM	0.701	0.768	0.896
($\sigma = 25$)	PSNR	25.18	26.33	31.71

Table 3: Quantitative comparisons of N2G and its variants.

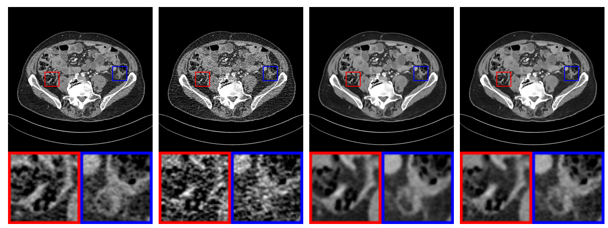
We design two variants of N2G. One is called N2G^D, which cancels detail destruction. The other is called N2G^U, which cancels the update delay. These networks are trained and tested on the dataset of Gaussian noise, and the results are reported in Table 3. Without detail destruction, the noisy image x^s synthesized by N2G is unrealistic, which leads to poor performance of the noise removal module. On the other hand, due to the cancellation of update delay, the gradient similarity loss $\mathcal{L}_{grad}$ has a negative effect on the noise removal module, thus damaging the denoising results. The combination of these two techniques can achieve excellent results of noise removal and noise extraction.

3.3 Medical Image Denoising

Computed tomography (CT) provides critical clinical information, but there are potential risks induced by X-ray radiation, such as cancer diseases and genetic damages. Given these risks, reducing the radiation dose as much as possible has become a trend in CT-related research. However, the reduction of radiation dose is associated with the increase of noise and artifacts in the reconstructed image, which may adversely affect subsequent diagnosis. Here, we show the application of Noise2Grad on low-dose CT denoising.

The training dataset is an authorized clinical low-dose CT dataset, which was used for the 2016 NIH-AAPM-Mayo Clinic LDCT Grand Challenge.[2] This dataset contains 5946 pairs of images with a slice thickness of 1mm. We divide them into 3 parts, 500 pairs as the test set, 2718 pairs as D^{noise}, and the remaining 2718 pairs as D^{clean}. N2G is compared with the fully-supervised U-Net. Results are shown in Figure 7 and Table 4. For low-dose CT denoising, the denoising performance of our N2G is close to the fully supervised U-Net. Our denoising network cleanly removes noise and restores high-quality images.

[2]https://www.aapm.org/GrandChallenge/LowDoseCT/



(a) Clean SSIM, PSNR (b) Noisy 0.772, 20.58 (c) U-Net 0.839, 26.95 (d) N2G 0.835, 26.51

Figure 7: Low-dose CT denoising example.

	Noisy	UNet	N2G
SSIM	0.752	0.828	0.816
PSNR	21.51	28.04	26.80

Table 4: Quantitative comparisons of N2G and its variants.

4 Conclusion

We proposed Noise2Grad, a novel algorithm that uses unpaired noisy and clean images to train denoising CNNs. The core idea of N2G is to extract noise from noisy images, and then superimpose it on other clean images to synthesize paired training data. To facilitate noise extraction, we use a noise removal module to eliminate the interference of the image background. Constrained by the image gradient, the noise removal module and the noise approximation module cooperate to extract noise finely. We demonstrate the effectiveness and wide applicability of Noise2Grad over multiple denoising tasks. Since Noise2Grad does not require precollected paired data and assumptions about noise statistics, it is a promising solution for many practical applications.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (No.2020YFC2003900), in part by the National Natural Science Foundation of China under Grants U19B2031, 61971369, in part by Hong Kong Research Grants Council through Research Impact Fund under Grant R-5001-18.

References

[Anwar and Barnes, 2019] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, 2019.

[Batson and Royer, 2019] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *ICML*, 2019.

[Buades *et al.*, 2005] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

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

[Dabov *et al.*, 2007] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[Du *et al.*, 2020] Wenchao Du, Hu Chen, and Hongyu Yang. Learning invariant representation for unsupervised image restoration. In *CVPR*, 2020.

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

[Hou *et al.*, 2020] Yingkun Hou, Jun Xu, Mingxia Liu, Guanghai Liu, Li Liu, Fan Zhu, and Ling Shao. Nlh: A blind pixel-level non-local method for real-world image denoising. *IEEE Transactions on Image Processing*, 29:5121–5135, 2020.

[Kaneko and Harada, 2020] Takuhiro Kaneko and Tatsuya Harada. Noise robust generative adversarial networks. In *CVPR*, 2020.

[Krull *et al.*, 2019] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *CVPR*, 2019.

[Laine *et al.*, 2019] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *NeurIPS*, 2019.

[Lehtinen *et al.*, 2018] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *ICML*, 2018.

[Liu *et al.*, 2018] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018.

[Liu *et al.*, 2019] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP*, 2019.

[Ma *et al.*, 2016] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

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

[Meng and De La Torre, 2013] Deyu Meng and Fernando De La Torre. Robust matrix factorization with unknown noise. In *ICCV*, 2013.

[Quan *et al.*, 2020] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *CVPR*, 2020.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[Selesnick, 2017] Ivan Selesnick. Total variation denoising via the moreau envelope. *IEEE Signal Processing Letters*, 24(2):216–220, 2017.

[Simoncelli and Adelson, 1996] Eero P Simoncelli and Edward H Adelson. Noise removal via bayesian wavelet coring. In *ICIP*, 1996.

[Wu *et al.*, 2020] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *ECCV*, 2020.

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

[Yan *et al.*, 2020] Hanshu Yan, Xuan Chen, Vincent YF Tan, Wenhan Yang, Joe Wu, and Jiashi Feng. Unsupervised image noise modeling with self-consistent gan. In *CVPR*, 2020.

[Yue *et al.*, 2019] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019.

[Yue *et al.*, 2020] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020.

[Zhang *et al.*, 2017] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[Zhang *et al.*, 2018] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[Zhao *et al.*, 2014] Qian Zhao, Deyu Meng, Zongben Xu, Wangmeng Zuo, and Lei Zhang. Robust principal component analysis with complex noise. In *ICML*, 2014.

[Zhu *et al.*, 2016] Fengyuan Zhu, Guangyong Chen, and Pheng-Ann Heng. From noise modeling to blind image denoising. In *CVPR*, 2016.