Paper Reading and Literature Review

Li Hantao, G2101725H, MSAI, hli038@e.ntu.edu.sg

With the advance of deep learning, we have witnessed significant progress in a wide spectrum of computer vision techniques in image segmentation (e.g. semantic segmentation, instance segmentation, panoptic segmentation), image classification, object detection (e.g. two-stage object detection, one-stage object detection, anchor-free object detection, few-shot object detection, etc.), image generation (e.g. image synthesis from noise, image composition, image-to-image translation, image editing). To address various challenges in these computer vision problems, we also witnessed the fast development of machine learning techniques in supervised learning, semi-supervised learning, weakly supervised learning, self-supervised learning, few-shot learning, unsupervised learning, transfer learning, unsupervised domain adaptation, etc.

This direct reading aims to equip you with capabilities of reading scientific papers in computer vision.

You are expected to select one paper of the above listed topics (or beyond the list as far as the paper is related to computer vision), and produce a paper reading report that should at least address the following points:

1. **what is the work about?**
2. **what are gaps of prior research works?**
3. **what are motivations of the performed research?**
4. **how does the proposed technique address the gaps?**
5. **what evaluation metrics were adopted to validate the designs?**
6. **what are constraints of the proposed technique?**
7. **what are possible future works?**
8. **……**

I. Introduction

Before I reached Singapore to study as a graduate student in MSAI, I worked as a semi-professional photographer for a while, and denoise is an essential step in post-processing landscape photography.

Therefore, I selected a paper in the field of image denoise published in CVPR2021, in which proposed the self-supervised ***Neighbor2Neighbor*** model [1] for concentrated reading. In the process of reading, it is inevitable to understand the basis of Neighbor2Neighbor, ***Noise2Noise*** [2], and articles related to the concept and modeling of image noise. This report will summarize the reading process and put forward some ideas in photography denoise at the end of the paper.

II. Background

In recent years, with the growth of deep neural network learning, almost all the newly proposed denoising models are based on it. In the classical neural network model, we need numerous image pairs (x_i, y_i) as the input training data, where x_i is the image with noise and y_i is the clean image, that is, the ground truth of x_i.

Nevertheless, it is challenging to obtain such a large number of real image pairs. If we need to capture such image pairs truly, we must continuously take multiple photos with short shutter time and high ISO, on the premise that the subject is static and the ambient light is unchanged. Then, one of them is used as x_i, and the average value of all images is y_i. However, the disadvantages of this method are also obvious: the premise of subjects' standstill makes it improbable to shoot fast-moving objects, such as sports games, vehicles, or streets. Thus, it is difficult to obtain the clean images y_i of moving items, even without considering the operation complexity. Paradoxically, these ambulant objects are the ones that demand denoising, while a still object can get a picture with a high signal-to-noise ratio through a long exposure time.

Instead of collecting real image pairs, researchers usually obtain (x_i, y_i) by artificially forging noise-clean image pairs recently. Directly adding a specific noise distribution is the simplest while not agreeing with the actual situation. Some scholars put forward the reverse procedure based on inverse ISP: Brooks et al. proposed the Unprocessing (UPI) ISP by adding noise using reverse model training [3]; Zamir et al. proposed a cycle ISP method, using a large number of sRGB images for training [4]. The restoration effect for authentic noisy images is significantly better than adding noise directly; however, there is still a gap with the actual noise. For example, UPI only selects shot noise and read noise, approximated to Gaussian distribution, for noise restoration.

Therefore, it is still challenging to obtain abundant clean-noise image pairs quickly and reasonably. In this regard, the Noise2Noise (N2N) model proposes that we can "learn to turn bad images into good images by only looking at bad images," that is, only images with noise are used to train the denoising network.

III. NOISE2NOISE MODEL

A. Basic Methods

Firstly, we start with the theoretical derivation of the N2N model. The conventional denoising method generally takes the noisy picture x_i as the input and the clear picture y_i as the output to train the network. On this basis, the neural network is trained to fit the mapping between the image pair to realize the denoising function. In the sample image pair (x_i, y_i), we minimize the empirical risk of (1).

$$\underset{\theta}{\operatorname{argmin}} \sum_i L(f_\theta(x_i), y_i) \tag{1}$$

which means, for task (2), the minimum value will be obtained in formula (3):

$$\operatorname{argmin} \mathbb{E}_y\{L(z, y)\} \tag{2}$$

$$z = \mathbb{E}_y\{y\} \tag{3}$$

Similarly, when *L1-loss* gets the minimum value, it is obtained at the median of the target. Such one-to-many regression task can be written as follows by neural network fitting:

$$\underset{\theta}{\operatorname{argmin}} \mathbb{E}_{(x,y)}\{L(f_\theta(x), y)\} \tag{4}$$

The above formula optimizes both x and y simultaneously. Suppose the inputs are independent of each other. We can apply a simple function to represent the scalar output while the task degenerates to formula (2). The complete training task can be decomposed into minimizing each training sample in (5).

$$\underset{\theta}{\operatorname{argmin}} \mathbb{E}_x\{\mathbb{E}_{y|x}\{L(f_\theta(x), y)\}\} \tag{5}$$

In the high-resolution circumstance, the correspondence from low-resolution to the high-resolution image is one-to-many, which means multiple corresponding to high-resolution images. If the network directly applies *L2-loss* to regress the high-resolution results, which tends to regress the mean value of the possible corresponding high-resolution images, so the predicted high-resolution images tend to be blurred.

When training the neural network with *L2-loss*, we can replace the fitting data with whose expected value is equal to the fitting object, while remaining the effect of the neural network. Hence, if the input data satisfying the conditional distribution is replaced by any distribution containing the same expected value, the parameter theta in (5) will not change. In other words, the network training will not be affected if we attach zero-means noise distribution to y_i. Eventually, we get the following optimization function as the same as (1).

$$\mathbb{E}_{y_i|x_i} = y_i \tag{6}$$

At this time, x_i and y_i suffered from the distribution disturbed of noise (do not have to be the same distribution). In addition, it is required (6) that two images should have the equivalent ground truth, while in most image denoising tasks, the expectation of the input image, the noise image, is the clean image itself.

To sum up, if we use two *independent, identically distributed, and zero-mean noise images* with the same ground truth, which means the same clean image expectation, the trained model has the same parameters as the clean-noise image pairs.

B. Evaluation

The author verifies the model through series of experiments. First, the author examines the noise distributed by Gaussian, Poisson, and Bernoulli. Then, they analyze the Monte Carlo image synthesis noise and MRI images. The results are outstanding, and some examples are shown in Fig. 1. Since this report focuses on Neighbor2Neighbor, details will not be discussed.

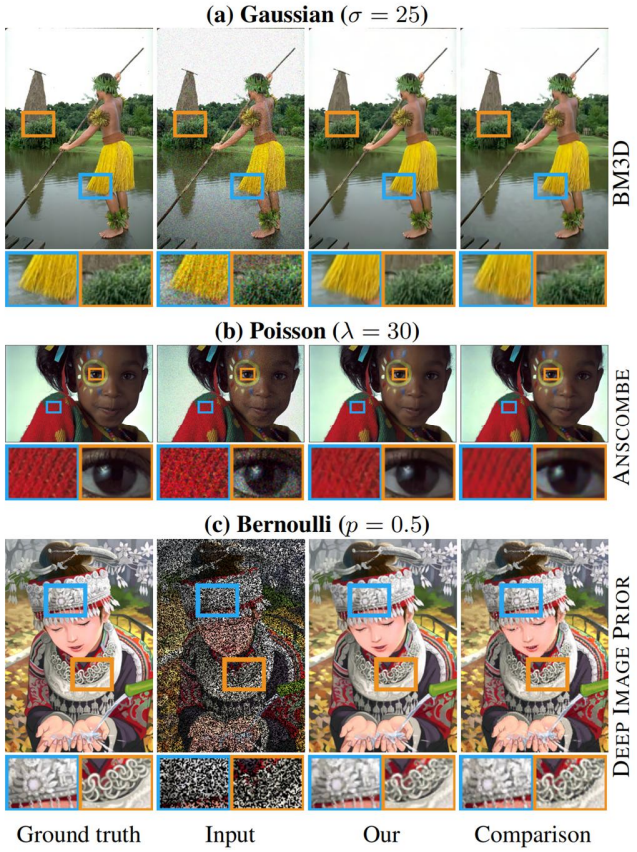


Fig. 1. Examples of N2N denoise model. [2]

The advantages of N2N are apparent: firstly, this method can train the denoise network without clear images, which is significant in many scenes. Secondly, noise can be removed without modeling the specific distribution, and it also has improvements compared with the high score paper Ambient-GAN (need to have a method to generate noise) of ICLR18 [5].

Nevertheless, at the same time, the limitations of N2N are also obvious: it is still complicated to obtain a pair of images that have independent and zero-mean distributed noise, while with identical ground truth. The most straightforward idea is to take two noisy shots of the same static scene, but the

contradiction is embarrassing: if we can take multiple noisy pictures with precisely the same ground truth, it is not challenging to obtain clean images. For scenes where we cannot obtain clean images, it is also difficult to obtain such noisy image pairs.

Therefore, it is vital to utilize only one noise image to train, which is the improvement proposed by Neighbor2Neighbor.

IV. NEIGHBOR2NRIGHBOR MODEL

A. Basic Methods

Let us summarize the essential idea of N2N in simple language. For a clean image x, we have its corresponding noisy image y, z, in which the noise is independent, identically distributed, and zero-mean. They have the same ground truth, clean image x, that is shown in (7). Then, for a network, the difference between the x and $f_\theta(y)$ has only a constant as the difference between z and $f_\theta(y)$, which is shown in (8).

$$\mathbb{E}_{y|x}(\mathrm{y}) = \mathbb{E}_{z|x}(\mathrm{z}) = x \tag{7}$$

$$\mathbb{E}_{x,y}\|f_\theta(y) - x\|_2^2 = \mathbb{E}_{x,y,z}\|f_\theta(y) - z\|_2^2 - const \tag{8}$$

where f_θ is the denoising network parameterized by θ.

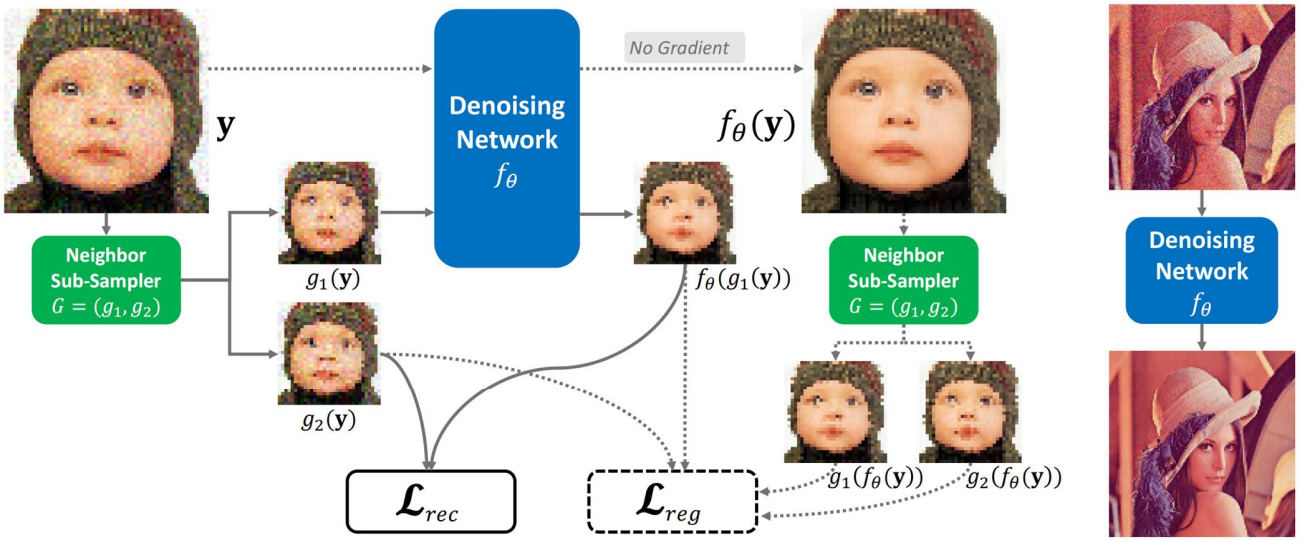
Neighbor2Neighbor, on this principle, further consider that if there is a little gap $\boldsymbol{\varepsilon}$ in ground truth of y and z, i.e

$$\varepsilon = \mathbb{E}_{y|x}(\mathrm{y}) - \mathbb{E}_{z|x}(\mathrm{z}) \neq 0 \tag{9}$$

At this time, substitute it into (8) to calculate the expectation, and then:

$$\mathbb{E}_{x,y}\|f_\theta(y) - x\|_2^2 = \mathbb{E}_{x,y,z}\|f_\theta(y) - z\|_2^2 - \sigma_z^2 + 2\varepsilon\mathbb{E}_{x,y}(f_\theta(y) - x) \tag{10}$$

When $\varepsilon = 0$, σ^2 will be the constant *const* in (8), where the formula becomes N2N's case. Furthermore, while ε approaches 0, this formula is the reasonable approximate of (8) in this state.



(a) Training (b) Inference

Fig. 2. Overview of proposed Neighbor2Neighbor framework. [1]
(a) Complete view of the training scheme. (b) Inference using the trained denoising network.

B. Image Pairs

How can we prepare two noisy images with ground truth of difference ε from a noisy picture, then put it into the network training as input data? The author proposes a method of neighbor sub-sampler. As shown in Fig. 3, the sampler traverses each 2x2 pixel block in the original image, randomly selects a pile of pixel pairs in each block, and then randomly assigns them to two sub-images so as to obtain image pairs with similar ground truth.

In another section, the author shows that if fixed pixel pairs are used in the downsampling process instead of random selection, reducing the data diversity and randomness, the outcome is slightly worse than random selection.

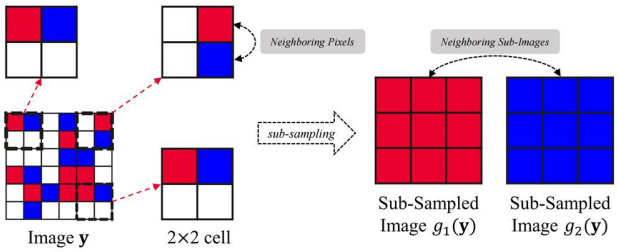


Fig. 3. The neighbor sub-sampler in Neighbor2Neighbor. [1]

C. Regularizer

In the preliminary training, the author observed that if the above loss function is directly applied for training, the output

result will be too smooth, for there have significant distinctions between the two sub-images passing through the sampler at the edge. Therefore, the author appends the property that an ideal denoising model should hold, that is, the order of denoising and sampling does not affect the denoising results, as a regular term to alleviate the training process. In other words, the author establishes a regularizer with the weight of γ to minimize the variation between the output of denoising before sampling and sampling before denoising, i.e

$$\min_{\theta} \begin{array}{c} \mathbb{E}_{x,y}\left\|f_\theta(g_1(y)) - g_2(y)\right\|_2^2 + \\ \gamma\mathbb{E}_{x,y}\left\|f_\theta(g_1(y)) - g_2(y) - (g_1(f_\theta(y)) - g_2(f_\theta(y))\right\|_2^2 \end{array} \quad (11)$$

where $g_i(y)$ is the i^{th} sub-image passing through the sub-sampler. The author takes this loss function as the final parameter to train the denoising model. The detailed process diagram has been shown in Fig. 2.In another section, the author analyzes the influence of weight on the training effect. When weight γ = 0, the image is too smooth; with the increase of weight, the details of the image are retained; when the weight is too large, the image will retain the noise, while the denoising effect becomes unsatisfying. Considering comprehensively, the author proposes that the effect is the best when γ = 2.

Noise Type	Method	KODAK	BSD300	SET14
Gaussian $\sigma = 25$	Baseline, N2C [26]	32.43/0.884	31.05/0.879	31.40/0.869
	Baseline, N2N [17]	32.41/0.884	31.04/0.878	31.37/0.868
	CBM3D [7]	31.87/0.868	30.48/0.861	30.88/0.854
	DIP [29]	27.20/0.720	26.38/0.708	27.16/0.758
	Self2Self [25]	31.28/0.864	29.86/0.849	30.08/0.839
	N2V [13]	30.32/0.821	29.34/0.824	28.84/0.802
	Laine19-mu [15]	30.62/0.840	28.62/0.803	29.93/0.830
	Laine19-pme [15]	**32.40/0.883**	**30.99/0.877**	**31.36/0.866**
	Noisier2Noise [22]	30.70/0.845	29.32/0.833	29.64/0.832
	DBSN [30]	31.64/0.856	29.80/0.839	30.63/0.846
	Ours	32.08/0.879	30.79/0.873	31.09/0.864
Gaussian $\sigma \in [5, 50]$	Baseline, N2C [26]	32.51/0.875	31.07/0.866	31.41/0.863
	Baseline, N2N [17]	32.50/0.875	31.07/0.866	31.39/0.863
	CBM3D [7]	32.02/0.860	30.56/0.847	30.94/0.849
	DIP [29]	26.97/0.713	25.89/0.687	26.61/0.738
	Self2Self [25]	31.37/0.860	29.87/0.841	29.97/0.849
	N2V [13]	30.44/0.806	29.31/0.801	29.01/0.792
	Laine19-mu [15]	30.52/0.833	28.43/0.794	29.71/0.822
	Laine19-pme [15]	**32.40/0.870**	**30.95/0.861**	**31.21**/0.855
	DBSN [30]	30.38/0.826	28.34/0.788	29.49/0.814
	Ours	32.10/**0.870**	30.73/**0.861**	31.05/**0.858**
Poisson $\lambda = 30$	Baseline, N2C [26]	31.78/0.876	30.36/0.868	30.57/0.858
	Baseline, N2N [17]	31.77/0.876	30.35/0.868	30.56/0.857
	Anscombe [19]	30.53/0.856	29.18/0.842	29.44/0.837
	DIP [29]	27.01/0.716	26.07/0.698	26.58/0.739
	Self2Self [25]	30.31/0.857	28.93/0.840	28.84/0.839
	N2V [13]	28.90/0.788	28.46/0.798	27.73/0.774
	Laine19-mu [15]	30.19/0.833	28.25/0.794	29.35/0.820
	Laine19-pme [15]	**31.67/0.874**	**30.25/0.866**	**30.47/0.855**
	DBSN [30]	30.07/0.827	28.19/0.790	29.16/0.814
	Ours	31.44/0.870	30.10/0.863	30.29/0.853
Poisson $\lambda \in [5, 50]$	Baseline, N2C [26]	31.19/0.861	29.79/0.848	30.02/0.842
	Baseline, N2N [17]	31.18/0.861	29.78/0.848	30.02/0.842
	Anscombe [19]	29.40/0.836	28.22/0.815	28.51/0.817
	DIP [29]	26.56/0.710	25.44/0.671	25.72/0.683
	Self2Self [25]	29.06/0.834	28.15/0.817	28.83/0.841
	N2V [13]	28.78/0.758	27.92/0.766	27.43/0.745
	Laine19-mu [15]	29.76/0.820	27.89/0.778	28.94/0.808
	Laine19-pme [15]	**30.88**/0.850	**29.57**/0.841	28.65/0.785
	DBSN [30]	29.60/0.811	27.81/0.771	28.72/0.800
	Ours	30.86/**0.855**	29.54/**0.843**	**29.79/0.838**

Table. 1 PSNR(dB)/SSIM of methods for Gaussian/Poisson noise. [1]

D. *Evaluation*

The first experiment is carried out on the synthetic RGB data set (with Gaussian noise or Poisson noise). As shown in Table.1, it is about 0.3dB lower in PSNR value than supervised denoising (N2C) and N2N. The performance is unmistakably better than other self-monitoring methods, and the performance is comparable to NVIDIA laine19, while it needs to estimate the parameters of the noise model in advance.

The second experiment is the denoise of real scene raw image (SIDD data set). Compared with N2C, the PSNR value is 0.1dB lower. Nevertheless, the performance is better than other self-monitoring methods. Furthermore, it is closer to the result of N2N/N2C than the synthetic images. At the same time, if a better network (RRG) is utilized, the performance will be significantly improved.

E. *Further Development*

In this paper, the author suggests that *"extend the proposed method to the case of spatially correlated noise and extremely dark images."*

From my point of view, firstly, we can enhance a better sampling strategy, while whether the current sampling strategy can gain the x and x+ε is still an intricacy. Secondly, we can attempt a more advanced denoising network. From the effect evaluation part, we can notice that the quality of the network model has a significant impact on the results. If we can find a more suitable network model, the denoising result can be further enhanced.

Finally, we should reconsider whether the noise model is reasonable. In the reasoning process of the two papers, noise is regarded as *"independent, identically distributed, and zero mean,"* but the rationality of this simplification remains to be discussed. In my opinion, this is also the primary reason for the poor performance of this model when *"in the case of spatially correlated noise and extremely dark images."* However, the Neighbor2Neighbor denoising ability of the denoising model for Poisson distribution is satisfactory, which benefits from the powerful ability of deep learning neural networks.

In the following parts, I will discuss the possible further improvement of denoising combined with my relevant experience in denoising in photography.

V. Further Discussion

To understand the noise information in detail, we refer to literature that presents in detail the characters of noise received by modern CMOS and CCD sensors and the modeling restoration of them [6]. Among it, all noise is classified into **photon shot noise**, the noise caused by the inherent physical properties of photons, and **read noise**, all noise generated between the photodiode and the output of the ADC circuit. (In some other literature, researchers often group dark current noise into a single category, but we think this classification method is unscientific and incomplete.) As shown in Fig.4, the noise ultimately affects the imaging quality includes *photon shot*

noise, dark current fixed pattern noise, KTC noise, source follower noise, offset fixed pattern noise, and *quantization noise*. Among them, we can ignore the noise caused by current-voltage conversion and digital conversion here (the third and fourth part), for it either has a zero-mean value that satisfies the constraints, such as *quantization noise* and *source follower noise*, or its impact is slight, for instance, the "stripe noise" caused by *offset fixed pattern noise* hardly can affect the imaging of the latest camera now.

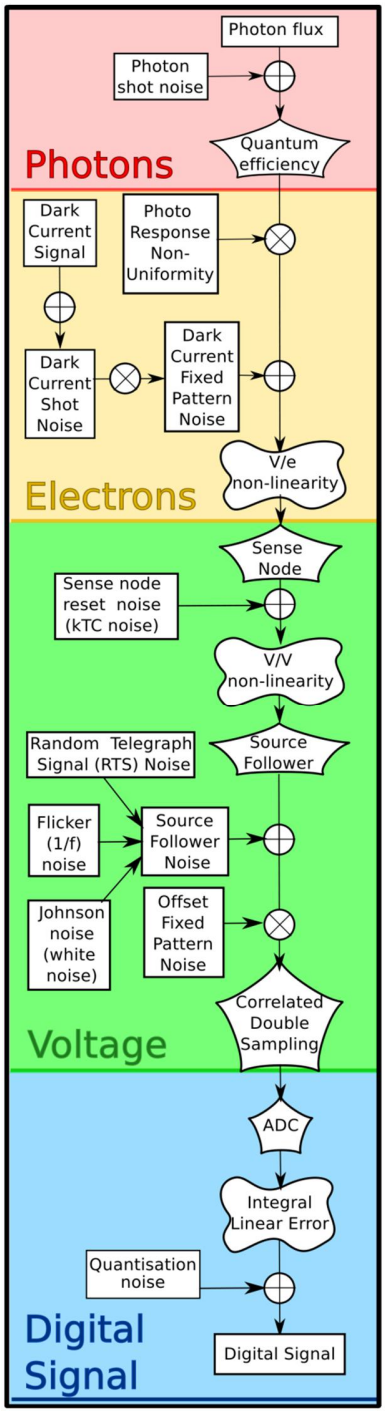


Fig. 4. The diagram of the photosensor model. [6]

Firstly, let us talk about *photon shot noise (PSN)*. In most cases, the noise we see all over the image is its contribution. This noise is caused by the uncertainty of the photon emission process. ***It conforms to all the definitions of Poisson distribution.*** The author emphasizes that it can be approximately expressed as Gaussian distribution only when the light intensity is tremendous, which is not feasible in low light. Hence, this should be the primary reason for the unsatisfactory outcome of the Neirgbor2Neighbor model on low light pictures mentioned by the author. The noise is summarized as "*independent, identically distributed, and zero-mean,*" which has a significant distinction from the photon shot noise distribution under low light, resulting in the poor effect of the denoising model in that circumstance.

Secondly, we will discuss the *dark current fixed pattern noise (DCFPN)*, which is the excess electrons generated by the heat generation of the silicon lattice inside the camera sensor, which accumulates over time. Researchers of the denoising model often ignore this kind of noise; however, it is the culprit affecting the image quality in the environment of long exposure time. It is mentioned that researchers often express the DCFPN approximation as Gaussian distribution (just like the N2N model), while this approximation has an awful result in fitting with the actual situation, which also leads to the poor consequence of the denoising model on DCFPN. Nevertheless, as "*a dark signal values much higher than the mean value of the dark signal,*" it significantly impacts image imaging. Now I will make a specific explanation in combination with the photos I took, shown in Fig.5, a night scene photo taken by myself. The enlarged part uses the raw file of the original photo. Firstly, it is apparent that DCFPN is distinctive from PSN, which is evenly distributed in the background, and the variation in saturation and brightness with the surrounding is obvious for DCFPN: it is a dazzling bright red spot here. If it is not removed, the film quality will be significantly diminished. Furthermore, although DCFPN is fitted as logarithmic distribution (or more complex pseudo-random distribution) in each scheming, the position of DCFPN is fixed in each power on-off workflow of the camera. In other words, we cannot eliminate DCFPN by taking multiple photos and taking an average/minimum value among them.

However, the current software that can denoise commercial photography cannot remove DCFPN, seriously affecting image quality. There are many commercial denoising models, such as Lightroom / Adobe Camera Raw (ACR), Noiseware, Skylum lunar, Nik Dfine 2, Photo Ninja, Near Image Pro, DxO Optics Pro, Topaz Denoise AI, etc. [7-14] The most broadly used is Adobe's ACR, wavelet noise reduction tool Nik Dfine 2 based on the frequency domain, and the Topaz denoise AI based on DL, which has sprung up in recent years. (Of course, we cannot find their specific models from Google Scholar.) Most of these models chiefly aim at the elimination of PSN. Taking Topaz as an example, its noise removal effect under low light has been satisfactory, while it is not satisfactory for the removal effect of DCFPN.

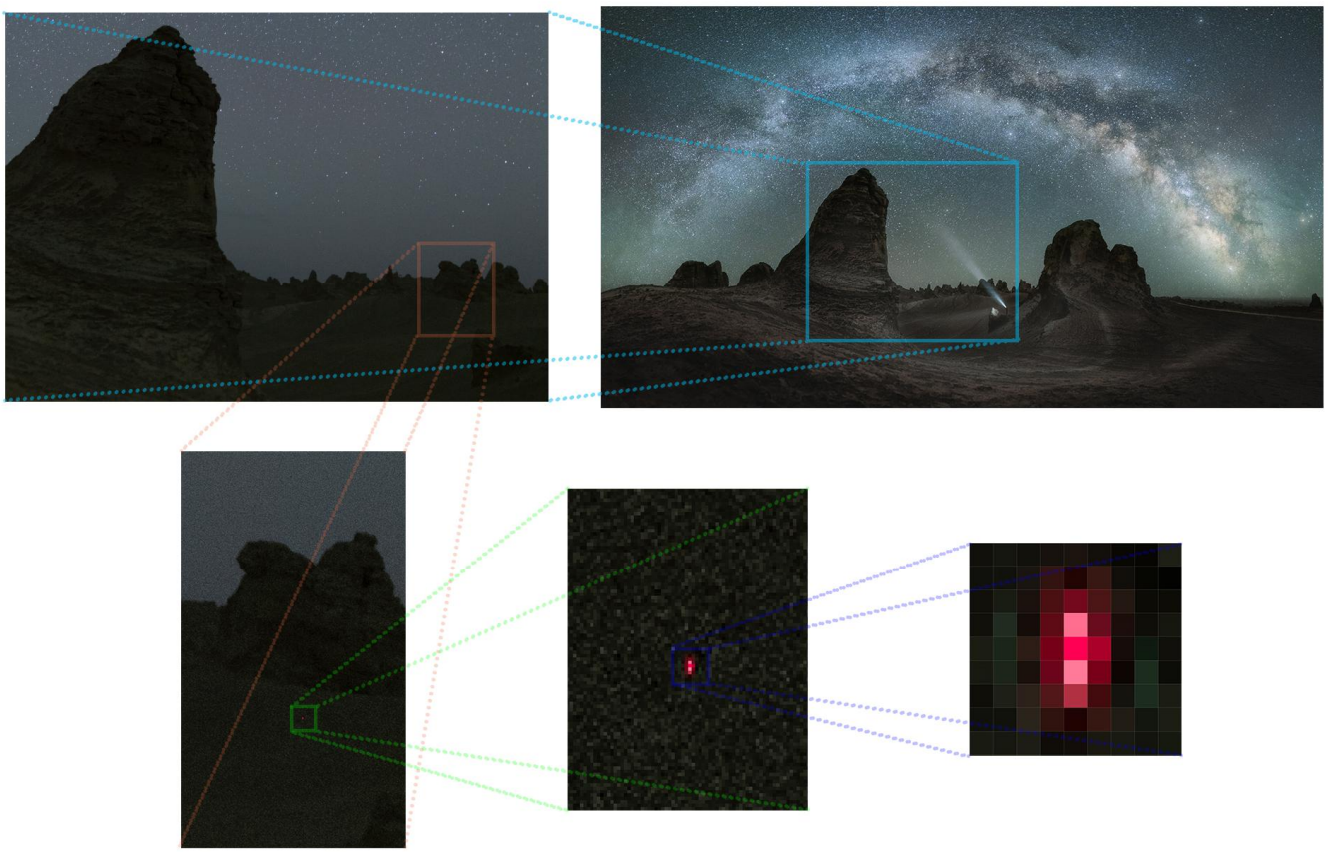


Fig. 5. One example of the DCFPN

Fig.6 shows the denoising effect of two denoising software on the DCFPN shown in Fig.5. We can notice that, even with the maximum denoising degree, where the mountain edge in the background has been completely blurred and distorted, the DCFPN has not been eliminated. This indicates that the model based on the frequency domain considered DCFPN as a piece of valuable information; even the denoising network model cannot effectively train DCFPN.

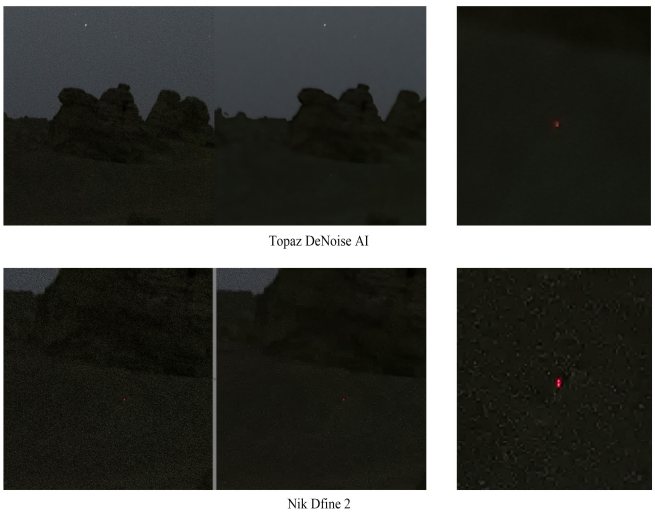


Fig. 6. Denoise result of Topaz and Nik

This reason can be analyzed: first, since DCFPN does not change its position in an on-off power workflow, if researchers do not pay special attention to the paradigm of collecting image pairs, the resulting clean-noise pair will take DCFPN as the ground truth, whether after the average of multiple photos or low-ISO long exposure. Secondly, the single noise "pixel" size of DCFPN is much larger than that of PSN. Since only one red noise is selected in the above figure, we supplement Fig.7 for further illustration. We can imagine that if using the 2x2 grid proposed by Neighbor2Neighbor for down sampling, no matter how random, the sub-images will contain this noise in the same position, which will also be recognized as the ground truth of the image. Consequently, the network trained with such input image will inevitably have poor denoising ability for DCFPN. In my opinion, this is the root cause of the poor effect of the model on spatially correlated noise mentioned by the author: the noise is simply summarized as "independent, identically distributed, and zero-mean," which is very different from the DCFPN distribution, resulting in the unsatisfactory result of the denoising model on spatially correlated noise.

Finally, we consider a question, that is, whether it is meaningful to investigate how to eliminate DCFPN, which is the noise that increases with exposure time and sensor heating, while its impact is limited viewing from the perspective of all photography categories. The camera manufacturer's control over DCFPN is also effective: the above photos are taken with

Canon R5. If you capture them with earlier cameras, such as 5D3, the appearance of DCFPN will be significantly greater. In addition, for the photography real suffers from DCFPN, such as deep space photography and astronomical photography, with ultra-long exposure time, the frozen CCD sensor is designed to reduce DCFPN for them.

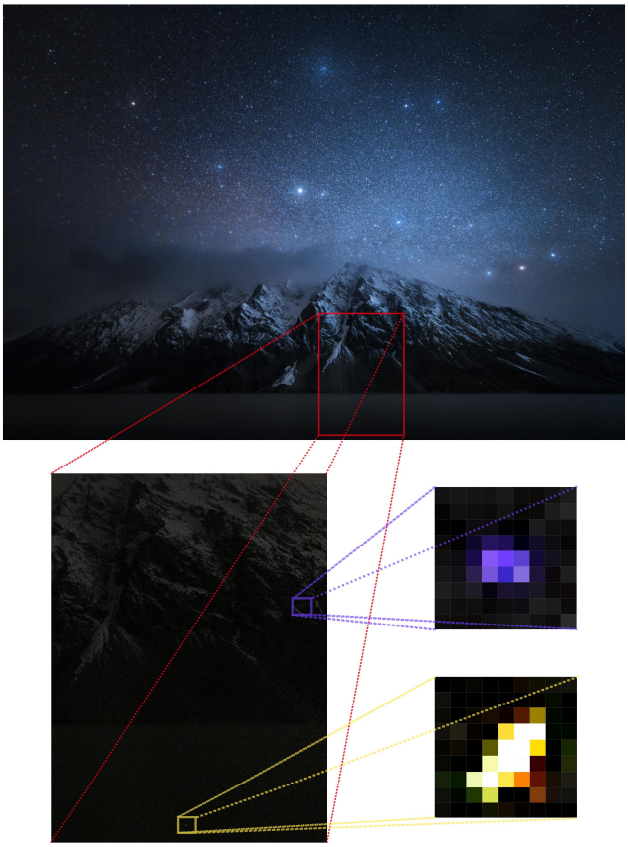


Fig.7. Another example of the DCFPN

However, we believe that the continuous miniaturization and lightweight of commercial cameras and sensors is inevitable, while there must be heating problems; hence, DCFPN can not be significantly eliminated for a long time. Even Canon's newly released R5 is still controversial because of the overheating when shooting video. Moreover, if the capacity of the denoising model for DCFPN can be developed, there will be room for designers to do trade-offs with the heating control. Therefore, consider improving the ability of existing denoising models to deal with DCFPN still has the prospect..

REFERENCES

[1] Huang, T., Li, S., Jia, X., Lu, H., & Liu, J. (2021). Neighbor2Neighbor: Self-Supervised Denoising from Single Noisy Images. ArXiv, abs/2101.02824.

[2] Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). Noise2noise: Learning image restoration without clean data. arXiv preprint arXiv:1803.04189.

[3] Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., & Barron, J. T. (2019). Unprocessing images for learned raw denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11036-11045).

[4] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M. H., & Shao, L. (2020). Cycleisp: Real image restoration via improved data synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2696-2705).

[5] Bora, A., Price, E., & Dimakis, A. (2018). AmbientGAN: Generative models from lossy measurements. ICLR.

[6] Konnik, M., & Welsh, J. (2014). High-level numerical simulations of noise in CCD and CMOS photosensors: review and tutorial. arXiv preprint arXiv:1412.4031.

[7] https://helpx.adobe.com/camera-raw/using/supported-cameras.html

[8] https://www.imagenomic.com/Products/Noiseware

[9] https://skylum.com/hans/luminar

[10] https://nikcollection.dxo.com/dfine/

[11] https://www.picturecode.com/

[12] https://ni.neatvideo.com/

[13] https://www.dxo.com/dxo-academy/

[14] https://www.topazlabs.com/denoise-ai