M.S. THESIS

Noise Modeling Generative Adversarial Network for Real-World Denoising

적대적 생성 신경망을 활용한 실영상 잡음 제거 기법

BY

GEONWOON JANG

FEBRUARY 2021

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Noise Modeling Generative Adversarial Network for Real-World Denoising

적대적 생성 신경망을 활용한 실영상 잡음 제거 기법

BY

GEONWOON JANG

FEBRUARY 2021

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Noise Modeling Generative Adversarial Network for Real-World Denoising

적대적 생성 신경망을 활용한 실영상 잡음 제거 기법

지도교수 이 경 무
이 논문을 공학석사 학위논문으로 제출함

2021년 2월

서울대학교 대학원

전기 · 정보공학부

장 건 운

장건운의 공학석사 학위 논문을 인준함

2021년 2월

위 원 장: 조 남 익 Namik Cho
부위원장: 이 경 무
위 원: 한 보 형

Abstract

Learning-based image denoising models have been bounded to situations where well-aligned noisy and clean images are given, or training samples can be synthesized from predetermined noise models. While recent generative methods introduce a methodology to accurately simulate the unknown distribution of real-world noise, several limitations still exist. The existing methods are restrained to the case that unrealistic assumptions are made, or the data of actual noise distribution is available. In a real situation, a noise generator should learn to simulate the general and complex noise distribution without using paired noisy and clean images. As a noise generator learned for the real situation tends to fail to express complex noise maps and fits to generate specific texture patterns, we propose an architecture designed to resolve this problem. Therefore, we introduce the C2N, a Clean-to-Noisy image generation framework, to imitate complex real-world noise without using any paired examples. Our C2N combined with a conventional denoising model outperforms existing unsupervised methods on a challenging real-world denoising benchmark by a large margin, validating the effectiveness of the proposed formulation. We also extend our method to a practical situation when there are several data constraints, an area not previously explored by the previous generative noise modeling methods.

keywords: Image Denoising, Image Restoration, Real-World Denoising, Generative Model, Generative Adversarial Network, Unsupervised Denoising

student number: 2018-27051

Contents

List of Tables

List of Figures

Chapter 1

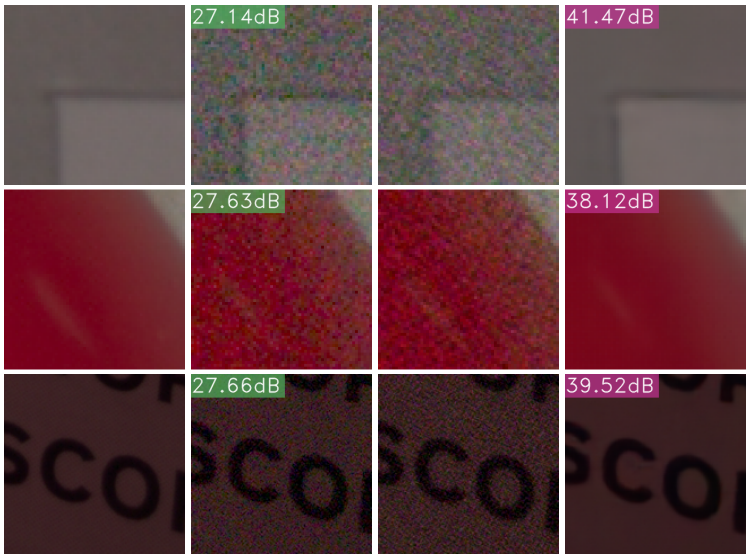
INTRODUCTION

Image denoising aims to remove unintended signals from a given noisy observation. The task has been considered as one of the fundamental vision problems and handled by numerous studies [10, 11, 12]. While recent deep convolutional neural networks (CNNs) have achieved promising performance [9, 13, 14, 15, 16] on the various dataset, several challenges prevent the methods from being used as practical applications. A primary limitation of the learning-based approaches is that they are usually data-driven, where an accurate fitting on training dataset does not guarantee the model to work well for real-world situations.

In the real-world scenario, the noise from a camera pipeline is different from the conventional assumption for ideal noise in several aspects. For instance, the widely-used additive white Gaussian noise (AWGN) formulation assumes that the term is signal-independent [17, 18], while real-world noises are not. Therefore, it is difficult to generalize a denoising algorithm toward real-world images when the model is trained on synthetic examples with the ideal noise assumption. As an alternative, few studies have collected well-aligned noisy and clean image pairs in the wild [19, 5] so that the following denoising methods can be trained in a supervised manner. While such an approach can be an effective way to deal with the real-world noise, it remains challenging to acquire the pairs due to practical issues. Recent self-supervised

deep approaches [18, 20] aim to deal with the limitations, but they usually leverage some statistical properties of the noise, which are insufficient to represent realistic noise [5, 21].

On the other hand, generation-based approaches [22, 23] have been proposed to deal with challenging real-world situations. Rather than using a single model, those methods usually adopt a two-stage pipeline for the denoising problem. First, a noise generator [24] is learned in an unsupervised fashion to simulate the distribution of given real noisy examples so that any clean images can be mapped to pseudo-noisy data. A denoising model can then be trained in a straightforward manner using the synthesized input and target pairs. Earlier methods [22, 17] train their generators using realistic noise maps extracted from nearly-plain image patches. However, the noise map is extracted with handcrafted operators, which may prevent the generalization ability toward the real-world camera noise. Several methods have provided more accurate supervision on the noise distribution [23] or trained their network with unpaired data [25] to generalize their generation models. Still, it has not been explored whether the generation-based formulation can deal with the situation where only the noisy examples are available, without sufficient clean images nor samples captured from the same scene.



(a) Clean (b) GT (c) C2N (d) C2N+DIDN

Figure 1.1: Examples of generated and denoised image from our proposed method. (a) Clean image, (b) Ground truth noisy image, (c) Generated noisy image from the proposed C2N, (d) Denoising results of DIDN [1] trained on the images generated by our C2N. These results shows that our C2N can accurately imitate the real noise.

In this paper, we introduce C2N, a novel generative noise modeling framework trained with no paired data. As shown in Figure 1.1, the C2N can learn a variety of complex noise distributions successfully and generates accurate noisy images from arbitrary clean images. We can combine any denoising models to this by training it on the generated pairs of pseudo-noisy and clean images to achieve state-of-the-art performance. Even with a significantly small amount of training data, our C2N can be trained to simulate the real-world noise and further preserve its performance under the constraints of lacking clean images. Our contributions is summarized as follows: (1) We propose a noise generator with explicit modules to express noise terms of according characteristics, making it possible to imitate a wide range of real-world noise and solve the structural problem of a noise-generating CNN. (2) With the data pairs generated

by our C2N, we train denoising models that outperform state-of-the-art unsupervised methods for denoising real photographs. (3) We show that our method performs well regardless of the data constraints in a practical situation.

Chapter 2

RELATED WORK

2.1 Deep Image Denoising.

After the DnCNN [9] model has achieved a significant performance gain over traditional methods [10, 11], CNN-based methods have become mainstream in the image denoising area. The FFDNet [13] model uses a noise level map as to remove spatially varying AWGN effectively. Using large models and complex architectures, the performance of denoising models can be improved by extracting rich features from the input noisy image [15, 26]. While such methods successfully erase AWGN with state-of-the-art performances, they still require numerous training pairs that contain the exact target noise distribution. Therefore, the real-world denoising task has remained challenging since the desired noise model is unknown without appropriate training examples.

2.2 Deep Denoising of Real-World Noise.

If enough amount of training samples are given, it is straightforward to train the methods mentioned to function properly. For such purpose, Anaya and Barbu [27] acquire the Renoir dataset where clean samples are synthesized from a sequence of low-ISO images. However, spatial misalignment and remaining noise in training pairs make

it challenging to use the dataset for practical purpose. The DND [21] dataset post-processes low-ISO images to align their spatial contents and illuminations with high-ISO counterparts. The SIDD [5] dataset captures noisy images under various lighting conditions with five different smartphone cameras. It gets clean images by merging the noisy images with complicated post-processing techniques to consider image alignments, lens motion, radial distortion, and optical stabilization. To deal with more challenging real-world scenarios, SID [28] and ELD [29] datasets introduce extreme low-light camera noise. With these real-world noise datasets, denoising models with various attention modules [30, 31], multi-scale resizing in features [32, 33, 1], or use of self-similarity in images [34, 35, 36] manage to remove complex real noises.

Nevertheless, it is difficult to collect large-scale real-world dataset for our specific purpose. To remove noise even when the accurate distribution of it is not given(bling denoising), the CBDNet [14] includes a part that performs noise level estimation based on Poisson-Gaussian Noise Model [37]. It is intended to also operate on real noisy images, receiving indirect supervision from training the denoiser. The Path-Restore [38] dynamically selects an appropriate restoration path for each region of an input image. The self-supervised denoising methods [18, 20] use only individual noisy images for training and estimates a pixel value of its input noisy image itself, where the value of that location is masked-out as 'blind spot'. Since these methods require the noise to satisfy strong statistical assumptions, [39, 40] modify these according to the prior knowledge of noisy images. Recently, the AINDNet [41] performs N2N [19] training with adaptation through varying noise levels. And the Noise2Blur [42] performs an additional procedure to preserve image details after training a model with blurred labels.

On the other hand, generation-based approaches [22, 23] have been proposed to deal with challenging real-world situations. Rather than using a single model, those methods usually adopt a two-stage pipeline for the denoising problem. First, a noise generator is learned in an unsupervised fashion [24] to simulate the distribution of

given real noisy examples so that any clean images can be mapped to pseudo-noisy data. A denoising model can then be trained in a straightforward manner using the synthesized input and target pairs. The GCBD [22] is the first generation-based approach for deep blind image denoising. Later, the GAN2GAN [17] method leverages better noisy-patch-extraction, generating more realistic noisy samples to train the following N2N model. While the methods above are limited to signal-independent and synthetic noise only, the Noise Flow [23] proposes a formulation to imitate challenging real-world noise. By leveraging the normalizing flow formed of invertible transforms, the shift of distribution between synthetic and desired noise maps can be learned precisely. Although the model successfully imitates in-camera noise occurring pipeline, true noisy and clean image pairs are required to get the correct noise distribution. Recently, the UIDNet [25] model handles the unpaired generative noise modeling with the sharpening technique for better noise separation. Furthermore, the NTGAN [43] method demonstrates that noise maps synthesized with a given camera response function (CRF) can be used for the following denoising network. Unlike the existing generation-based denoising methods, we introduce a novel noise generator that can be trained without any paired data or heuristics.

Chapter 3

C2N: Clean-to-Noisy Image Generation Framework

3.1 Complexity of Real-World Noise

To generate realistic noise with CNNs, it is necessary to understand the properties of real-world cameras and their statistical behavior. Due to several physical limitations, the noise occurs from various sources, including electronic sensors, in-camera amplifiers, photon noise, quantization, and compression artifacts [44]. Combining all these factors, the noise term n is mixed with an underlying clean signal x, resulting the noisy observation y as follows:

$$y = x + n. \tag{3.1}$$

In traditional deep denoising methods [9, 13], the noise term n is usually simplified as an ideal additive white Gaussian (AWGN), i.e., $n \sim \mathcal{N}(0, \sigma^2)$, where σ denotes the standard deviation. On the other hand, the photon noise is signal-dependent, where Poisson distribution can be used to simulate the case [37]. Considering the signal-independent and dependent terms together, the resulting Poisson-Gaussian noise model can be defined as follows:

$$n \sim \mathcal{N}(0, \sigma_s^2 x + \sigma_c^2), \tag{3.2}$$

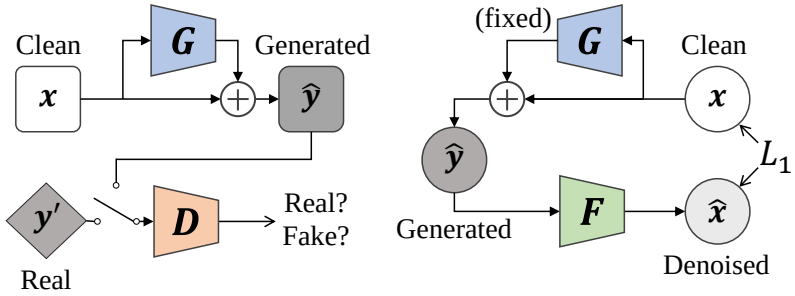
where σ_s^2 and σ_c^2 are tunable parameters. While the noise model in (3.2) can provide a proper approximation of the realistic noise [14, 43], several studies [44, 29] have

demonstrated that the real-world cases appear to be much more complicated. Furthermore, the physical limitations of in-camera electronic devices and in-camera compression pipeline make the noise term to exhibit random spatial pattern [45]. Such properties lead the noise term n and its local neighbors to be spatially correlated, making the precise modeling more challenging.

To deal with the problem without using paired data, the previous approaches [14, 43] construct noise maps using a synthetic noise model and known camera response function (CRF). However, such handcrafted features prevent those methods from being generalized toward realistic configurations. Therefore, we propose to adopt a learning-based method, namely C2N, to simulate the real-world noise rather than using some handcrafted formulations. Our framework fully utilizes the advantage of unsupervised learning to simulate the more comprehensive real-world noise, with novel design components and objective terms.

3.2 Learning to Generate Pseudo-Noisy Images

A denoising network F aims to reconstruct the underlying clean signal x from a given noisy observation y in (3.1). When enough training pairs are available, the model can be trained in a supervised manner to estimate the clean signal. However, in real-world scenarios, it is challenging to acquire *ideal* clean images even with complicated post-processing [5], and so as the well-aligned noisy-clean training pairs for supervised learning.



(a) Training generator (b) Denoising

Figure 3.1: Our two-step pipeline for real-world denoising. (a) Our method first learns to generate samples from target noise distribution, with a noise generator G. We used clean image x and noise image y' to be unpaired. (b) Secondly, using the generated pairs, we train a denoising model F. The clean images for this step can be sampled from an arbitrary distribution $P'_C \neq P_C$.

Hence we first train our generator in C2N framework for target noise distribution, and train the following denoising model F on the generated noise. Figure 3.1 shows the two-step pipeline of our method.

Our generator G is designed to synthesize a realistic noise map $\hat{n}$ for a given clean image x to produce the pseudo-noisy image $\hat{y}$ as follows:

$$\hat{y} = x + \hat{n}, \quad \hat{n} = G\left(x, r\right), \tag{3.3}$$

where r is a random vector to reflect stochastic behavior of the noise. We spatially replicate the 32-dim random vector r through all pixel positions of x, similar to the GAN applications [46, 47].

Simultaneously, we train a discriminator network D to distinguish whether a given noisy image is synthesized from our generator G or sampled from the real-world dataset. The two networks G and D can be optimized in an adversarial way [24],

using the Wasserstein distance [48] as follows:

$$\begin{aligned}\mathcal{L}_{\text{adv}}(D,G) = &\ \mathbb{E}_{y'\sim P_N}[D(y')] \\ &- \mathbb{E}_{x\sim P_C, r\sim P_r}[1 - D(x + G(x,r))] \\ &+ \lambda \mathbb{E}_{x_\delta \sim P_\delta}[(\|\nabla_{x_\delta} D(x_\delta)\|_2 - 1)^2],\end{aligned} \tag{3.4}$$

where the term is minimized with respect to G and maximized with respect to D. P_N and P_C denote the distribution of the real-world noisy and clean images, respectively. The real noisy sample y' is sampled from P_N, and we note that the corresponding clean image is not available in our configuration. P_r is a known distribution where the random vector r is drawn from, e.g., Gaussian. For stable learning, we adopt the gradient penalty [48], which is weighted by a hyperparameter $\lambda = 10$, where $x_\delta \sim P_\delta$ is a linear interpolation of the generated and real images.

The significant advantage of the proposed C2N framework is that the generator of it can synthesize realistic noise without adopting handcrafted features. However, the generated noise $\hat{n}$ may bias the image color and negatively affect the overall framework if our C2N is trained without any constraints. To deal with the case, we additionally define a stabilizing loss term $\mathcal{L}_{\text{stb}}$, which is defined as follows:

$$\mathcal{L}_{\text{stb}} = \sum_c \left\| \frac{1}{mHW} \sum_{k,h,w}^{m,H,W} \hat{n}_{k,h,w} \right\|_1, \tag{3.5}$$

where k is index of each images in a mini-batch of size m, h and w are index of pixel positions in height and width of an image of size H, W, respectively, and c is index of each color channels. By minimizing the stabilizing loss, the channel-wise average of the generated noise approaches zero, which prevents the color-shifting problem. Since we do not take a mean over a single sample or local area, the generated noise may have varying nonzero local means depending on the underlying signal to consider spatially-correlated patterns. Combining our two loss terms (3.4) and (3.5), we optimize the total loss $\mathcal{L}_G = \mathcal{L}_{\text{adv}} + w_{\text{stb}}\mathcal{L}_{\text{stb}}$ to train our C2N method, where $w_{\text{stb}} = 0.01$.

3.3 C2N Architecture

The previous generation-based approaches have been limited to signal-dependent [22] or spatially uncorrelated [23] noise terms, which limit their practical usages. Therefore, we define a new generator to represent diverse and complex noise distributions discussed in 3.1. Starting from the simplest CNN structure, we gradually implement several design components to express more general properties of real-world noise.

3.3.1 Signal-Independent Pixel-Wise Transforms

A general CNN is basically a sequence of fixed filtering operations. Therefore, the structure is not appropriate to synthesize purely noisy outputs but tends to generate some structured patterns. Instead, the noise generation can be formulated as a nonlinear mapping from an initial random noise s^I to the desired complex noise. A noise generator often consists of a sequence of transposed convolutions, , upconv, on a random vector [22], or pixel-wise operations applied to the noise map of known distribution [23]. Similarly, we construct a basic pixel-wise noise transformation module $G^I_{1\times1}$ to simulate spatially i.i.d. noise. Specifically, we have modified residual blocks [49] for low-level vision problems [3] with 1×1 convolutional layers. We sample the initial noise s^I from the standard normal distribution while using uniform distribution shows similar behavior.

3.3.2 Signal-Dependent Sampling and Transforms

To express signal-dependent noise, our noise generator should extract useful features from the clean input signal. However, it is not desirable for the generated noise map $\hat{n}$ to show too much correspondence with the given image and follow some structures in the scene. To effectively represent the signal-dependent term without biased to the given sample, we use convolutional features from the input as a statistics of the initial noise s^D. At the front of G^D, the signal-dependent part of G, we define the image

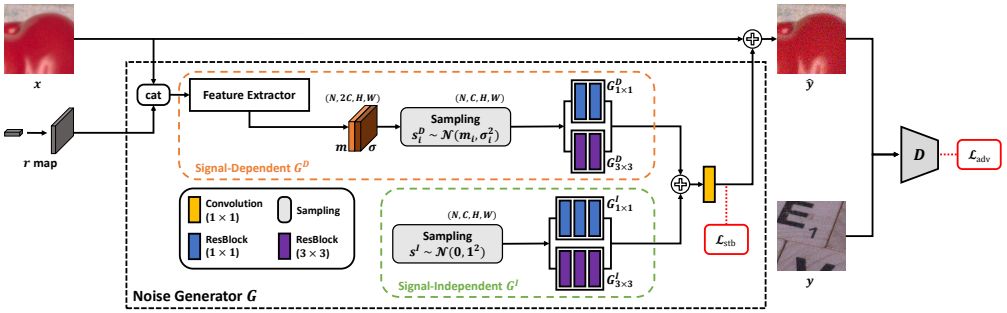


Figure 3.2: Overview of C2N framework. r map denotes the spatially replicated random vector r. For sampling a initial noise map for the signal-dependent transforms, we used reparameterization trick [2] to preserve gradients of the parameters.

feature extractor consisting of 5 ResBlocks with 3×3 convolutions. Its outputs feature map as the mean and standard deviation of each position-wise normal distributions. We then sample the initial signal-dependent component of noise s_i^D at each position i, and apply another pixel-wise noise transforms $G^D_{1\times1}$.

3.3.3 Spatially Correlated Transforms

A number of existing manual noise models [37] and denoising methods assume spatially uncorrelated noise. On the other hand, C2N handles color sRGB image containing various conversion and compression degradations from in-camera post-processing [45] in end-to-end manner. To achieve this, We add transforms of 3×3 convolution in G^I and G^D each, which are denoted as $G^I_{3\times3}$ and $G^D_{3\times3}$. These 3×3 convolutions tends to transform the noise term therefrom, rather than making unwanted artifacts that affect the content of the underlying signal.

Figure 3.2 shows the overall structure of our C2N framework. We set the number of channels C as 64, and all the intermediate features from s^I and s^D have the same number of channels. We add all the features transformed by $G^I_{1\times1}$, $G^I_{3\times3}$, $G^D_{1\times1}$, $G^D_{3\times3}$ into one, and take the last 1×1 convolution to reduce its number of channels according

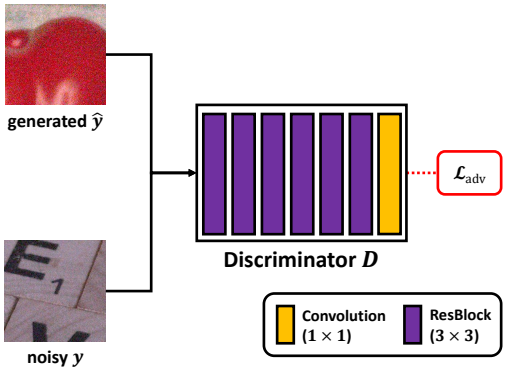


Figure 3.3: Discriminator architecture. The ResBlocks, like those of the generator, are modified for low-level vision problems [3].

to the color space.

3.3.4 Discriminator

We define the discriminator architecture as six sequential ResBlocks with 3×3 convolutions, and a 1×1 convolution applied to reduce the number of channels to one. Figure 3.3 shows the discriminator architecture of our C2N framework. The result values produced by the discriminator is averaged through pixel positions so that it can output one value for discriminate whether the image is real or generated one.

3.4 Learning to Denoise with the Generated Pairs

With the C2N, it is straightforward to optimize the following denoising network F. We first generate pseudo-noisy images $\hat{y}$ from the clean examples x and use the pairs to train a denoising model in a supervised manner [9, 13]. Similar to the previous deep denoising methods [1, 30], we minimize the L_1 reconstruction loss which is defined as follows:

$$\mathcal{L}_{\text{rec}} = \frac{1}{m} \sum_{k=1}^{m} \left\| F(\hat{y}) - x \right\|_1 , \tag{3.6}$$

where $\hat{y}$ is pseudo noisy image generated by $\hat{y} = x + G(x, r)$, on a clean image $x \sim P'_C$ sampled from arbitrary distribution of clean images.

The major advantages of our approach is that our framework is independent to the selection of following denoising architecture. Previous attempts like [17, 25] train their noise generator and denoising model jointly. Since the C2N model doesn't get any supervision from the reconstruction loss $\mathcal{L}_{\text{rec}}$, the generated images are not specialized for certain denoising model. In C2N, whether the generated sample $\hat{y}$ is good or not is determined with only resulting pseudo-noisy images. Also, in real situation, a denoising model cannot be trained with (3.6) because there are none or very few clean images available in the same scene distribution.

Chapter 4

Experiment

4.1 Experimental Setup

4.1.1 Dataset

To train and test our method on synthetic noise, we use clean color images from the BSD500 [7] dataset, which consists of 68 test samples (CBSD68) and 432 training images. We also use the high-resolution images from the DIV2K [6] dataset to synthesize images with more complex noise. To train our C2N and the following denoising model on challenging real-world noise, we leverage the SIDD [5] dataset captured under a realistic environment. Recent generation-based methods on real noisy images [23, 25] also use this dataset for training and evaluation, for it covers a wide range of capturing conditions and noise from them. The training split of the SIDD dataset (SIDD Medium) contains 320 noisy and clean image pairs, on which we train our C2N model by sampling each image in an unpaired manner. We evaluate the denoising models followed by the C2N with the SIDD Benchmark and its validation set.

4.1.2 Implementation Details and Optimization

To optimize our C2N framework, we crop 36,000 unpaired noisy and clean patches of 96×96 from each dataset unless mentioned otherwise. All training samples are augmented by random flipping and 90° rotations to construct a mini-batch of size 36. We used the Adam optimizer [50] with an initial learning rate of 10^{-4}, which is multiplied by 0.8 every 3 epochs. A single C2N model is trained over 36 epochs. After the whole training is done, we create pseudo-noisy images from a set of clean samples to train the following denoising model. We train our denoising model only with the generated pseudo-noisy and clean image pairs. Following the C2N pipeline, the denoising network is trained in a similar manner, where we adopt 36,000 generated pairs of 96×96 image patches with a mini-batch size of 16. The Adam [50] optimizer is used with an initial learning rate of 10^{-4}, which is halved every 4 epochs for the total 16 epochs. For the denoising architecture, we choose CDnCNN-B [9] as our baseline unless otherwise stated.

We applied the self-ensemble technique [51, 3] for the final denoising results specified as so. For a noisy image, we make total 8 augmented inputs by flipping and 90° rotations, including the original. We obtain each corresponding results and convert them back to the original geometry by the inverse transformations. Finally, we average all the results to get the self-ensembled result.

4.2 Model Analysis

To confirm the validity and usefulness of our C2N framework, we first analyze how each of our design components supports the proposed method in handling the synthetic noise of various properties.

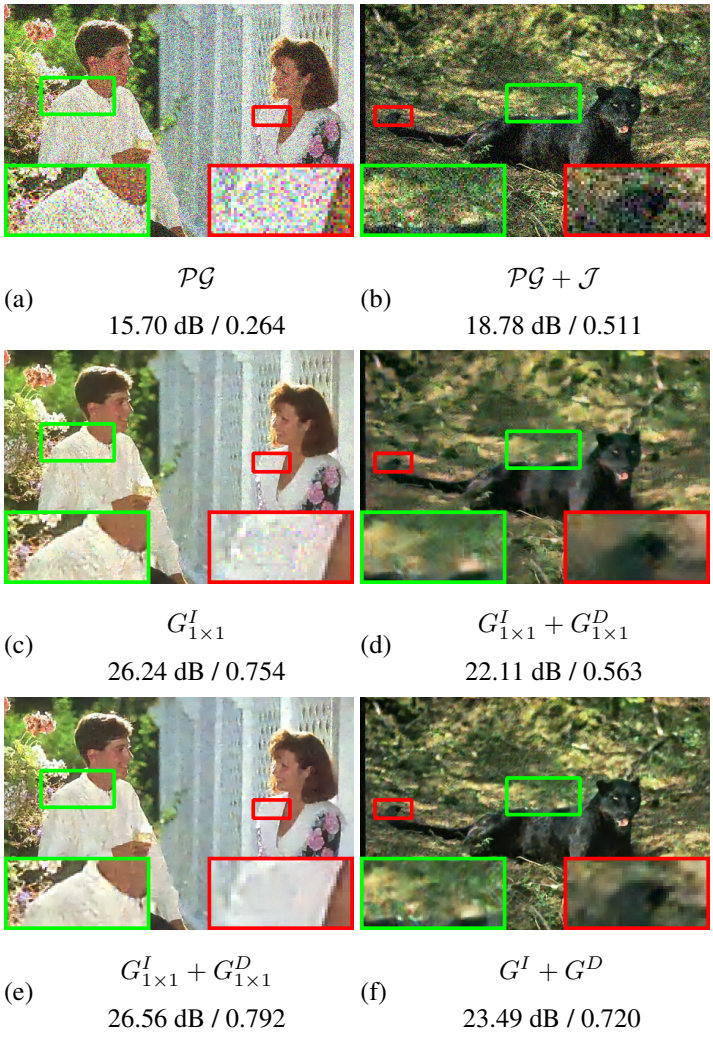


Figure 4.1: Ablation study on various synthetic noise. (a), (b) Synthetic noisy image of Poisson-Gaussian noise and the same degradation with JPEG compression, respectively, (c), (d), (e), (f) Results of our denoising models trained on the generated data from independent C2N frameworks. Values represent PSNR(dB) / SSIM [4] of each image with respect to the ground-truth.

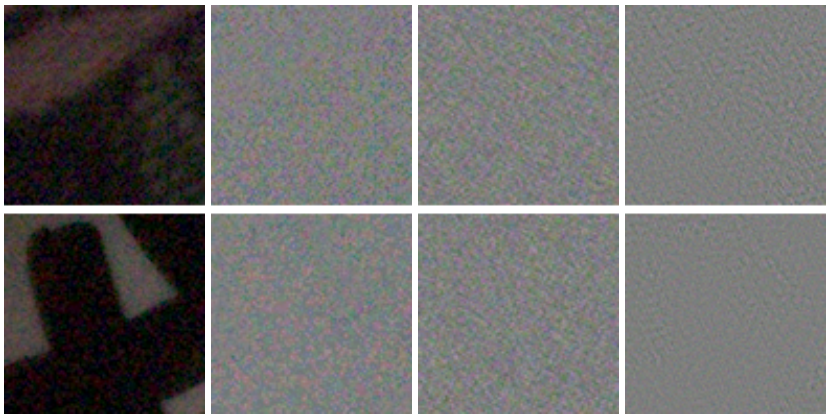
Figure 4.1 shows how each module in our C2N model can be used to handle the unique characteristic of complicated noise explicitly. Each C2N variant has a different range of expressions depending on the existence of each module. Here, $\mathcal{PG}$ and $\mathcal{PG} + \mathcal{J}$ stand for the Poisson-Gaussian noise of $(\sigma_c, \sigma_s) = (25, 50)$ and the same Poisson-Gaussian noise with JPEG compression of quality factor 70, respectively. We train the C2N models on high-resolution clean images from the DIV2K dataset with corresponding noisy examples in an unpaired setting.

Since the model only with $G^I_{1\times1}$ module is not proper to synthesize signal-dependent noise, the following denoising model does not remove Poisson-Gaussian noise well, especially on bright regions as shown in Figure 4.1(c). By explicitly considering the signal-dependent transformation module in our method, the following denoising method can reconstruct a visually pleasing result, as shown in Figure 4.1(e). Figure 4.1(d) illustrates that our C2N model cannot be used to imitate spatially-correlated noise with 1×1 convolutional layers only. Introducing 3×3 convolutions to the generator improves the representation power of our method and performance of the following denoising model on challenging JPEG artifacts removal, as shown in Figure 4.1(f).

Training Data		Test Noise Level $\sigma = 15$	$\sigma = 25$	$\sigma = 50$
Synthetic	$\sigma = 15$	**33.48**	27.24	18.30
	$\sigma = 25$	31.39	**30.68**	20.88
	$\sigma = 50$	27.82	27.81	**27.14**
C2N		**32.96**	**30.51**	**27.09**

Table 4.1: Denoising performance on synthetic AWGN. PSNR(dB) is calculated on the CBSD68 dataset. We note that the C2N models are trained for each noise level independently.

Besides, we evaluate our method on synthetic data and compare the performance with supervised denoising models, as shown in Table 4.1. The supervised models are trained with pairs of clean and noisy images synthesized by AWGN with the exact corresponding noise level. On the other hand, our method does not use the paired data obtained from the known noise model. Instead, the proposed C2N framework can learn to simulate various noise levels in an unsupervised fashion. Nevertheless, our approach shows comparable performance to the denoising models trained in a supervised manner, regardless of the underlying noise level.



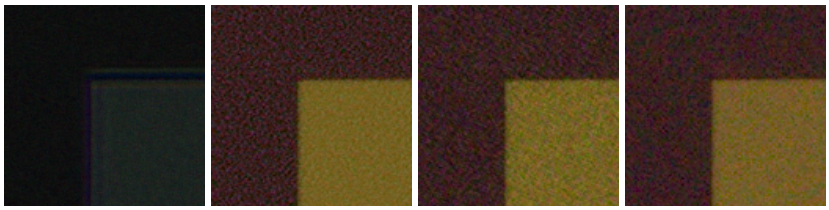
(a) GT x (b) GT n (c) Ours $\hat{n}$ (d) Baseline $\hat{n}$

Figure 4.2: Visual comparison of C2N and baseline generator. (a) Ground truth noisy images, (b) Ground truth noise maps, (c) Generated noise maps from the generator in C2N, (d) Generated noise maps from baseline generator.

Secondly, we check the effectiveness of each modules in our C2N framework under the real-world degradations. For comparison, we construct a network consisting of conventional 3×3 ResBlocks with the same number of parameters as the generator in C2N, and call it a baseline generator. The input of the baseline generator is a concatenation of the same clean image x, r map, and initial noise map s^I in Figure 3.2. After training our C2N and baseline generator on the the SIDD with the same strategy, we compare the noise maps generated by each model. The results in Figure 4.2 shows that the generator in our C2N can generate realistic noise maps, whereas the baseline generator tends to produce a particular texture. The stochastic behavior and explicit transforms of the C2N model help its output to maintain the characteristics of noise, solving the difficulties of a regular CNN functioning as a noise generator.

G^I	G^D	$G^I_{3\times3}, G^D_{3\times3}$	$\mathcal{L}_{\text{stb}}$	PSNR(dB)
✓	✓	✓	-	9.81
✓	-	-	✓	28.54
✓	✓	-	✓	31.74
✓	-	✓	✓	32.19
-	✓	✓	✓	32.21
✓	✓	✓	✓	**34.08**

Table 4.2: Model ablation study on SIDD validation set. The notation of each modules follows the Figure 3.2.



(a) wo $\mathcal{L}_{\text{stb}}$ (b) wo G^I (c) all (d) GT noise

Figure 4.3: Generated samples from model ablation study. (a) without stabilizing loss, (b) without independent transforms, (c) with all modules and stabilizing loss, (d) Ground truth noisy image.

Also, Table 4.2 and Figure 4.3 shows that each component of our C2N model is also essential for real-world noise modeling. Since our method learns the noise model distribution without any accurate noise maps or a heuristic technique to stabilize the learning, the stabilizing loss $\mathcal{L}_{\text{stb}}$ plays an important role. By comparing the 3rd to 6th rows of Table 4.2, we can again verify the necessity of G^I, G^D and 3×3 convolutional transforms. For example, the model without G^I of the 5th row lacks in expressing accurate noise and produces samples that differ in noise distribution, like the one in Figure 4.3. From the results, we confirm that the entire elements of the C2N together

can well simulate challenging real-world noise.

4.3 Results on Real-World Noise

Method		PSNR(dB)	SSIM
Non-learning -based	BM3D [11]	25.65	0.685
	WNNM [12]	25.78	0.809
	K-SVD [52]	26.88	0.842
	EPLL [53]	27.11	0.870
Supervised	TNRD [54]	24.73	0.643
	DnCNN [9]	28.46	0.784
	CBDNet [14]	33.28	0.868
Unsupervised	N2V [18]	27.68	0.668
	UIDNet-NS [25]	31.34	0.856
	UIDNet [25]	32.48	0.897
	C2N + DnCNN	**33.76**	**0.901**
	C2N* + DnCNN	**34.00**	**0.907**
	C2N + DIDN	35.02	0.932
	C2N* + DIDN	35.35	0.937

Table 4.3: Quantitative evaluation on the SIDD benchmark. We adopt the two-stage pipeline which is denoted by 'C2N + Denoiser.' * denotes the method with self-ensemble strategy.

Table 4.3 shows the performance of our method evaluated on the SIDD Benchmark of denoising in sRGB space. We categorize the compared methods as non-learning-based, supervised, and unsupervised methods. The supervised methods are trained with the paired data from the SIDD dataset. Though they are not on the same track

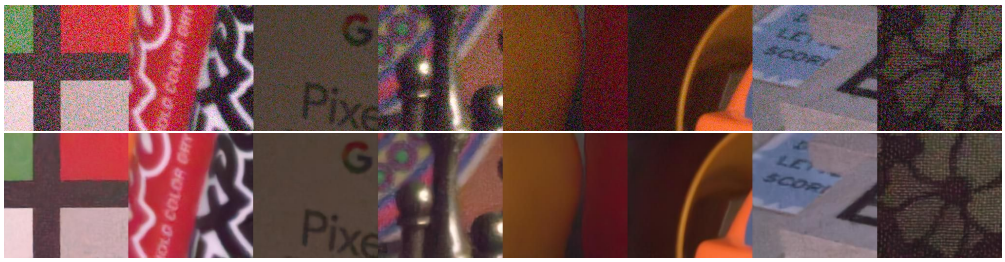
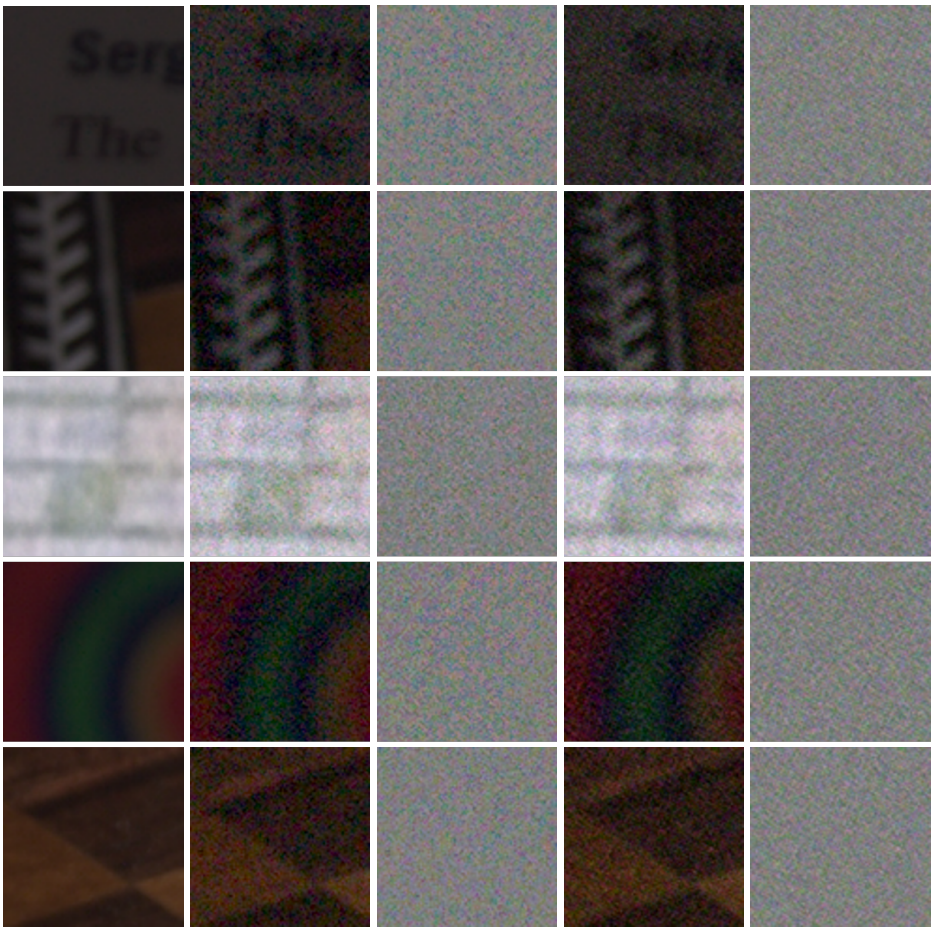


Figure 4.4: Qualitative results on the SIDD benchmark. The first row images are from the SIDD Benchmark, and the images of the second row are the denoising results of our method. Representative examples are selected by benchmark providers for the purpose of announcing the results.

of denoising methods learned without paired supervision, we note that our method outperforms them, including the CBDNet, a framework proposed for blind denoising of real photographs. The non-learning-based methods do not include the process of learning, and denoising is performed using the priors obtained from the test images. The unsupervised methods are trained without using exact data pairs of noisy and clean images in the SIDD.

We note that our method outperforms existing unsupervised methods by a large margin. Especially, our method with C2N and DnCNN denoiser shows better results than the UIDNet, a previous generative noise modeling method with its denoising backbone adopted from the same DnCNN model. We also use much fewer data than the 520,965 noisy and clean image patches of 64×64 size used by UIDNet. We further improve the performance of our method by using the DIDN [1] as its denoising model, which has larger capacity of $\sim$217M parameters compared to that of $\sim$0.67M parameters of the DnCNN. It shows that we can train arbitrary denoising models with C2N, unlike UIDNet with a fixed denoising backbone.



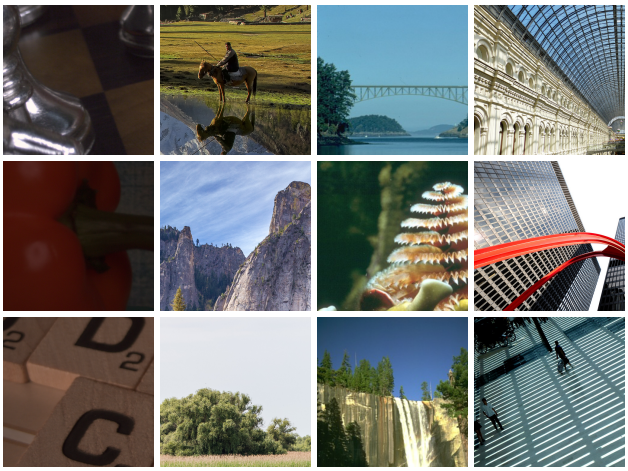
(a) Clean (b) GT (c) GT res (d) C2N (e) C2N res

Figure 4.5: Examples of ground truth noisy image and generated image from our C2N. (a) Clean image, (b) Ground truth noisy image and (c) its residual noise map, (d) Generated noisy image from the proposed C2N and (e) its residual noise map.

Figure 4.5 shows more pseudo-noisy samples generated by our C2N for visual comparison. The generated samples closely resembles the ground truth samples. In the residual noise maps of our C2N, there are no artifacts or loss of randomness that destroy the content of the signal.

Figure 4.4 shows the example noisy images of the SIDD benchmark and our denoising results on them. Along with quantitative results, our method is able to reconstruct clean images successfully without losing the details.

4.4 Performance Under Practical Data Constraints



(a) SIDD (b) DIV2K (c) BSD (d) Urban100

Figure 4.6: Different scenes in various datasets. Sampled clean image patches from (a) the SIDD [5], (b) the high-resolution images of the DIV2K [6], (c) the 432 training images of the BSD [7], (d) the high-resolution images of the Urban100 [8].

Due to several physical limitations [19], it is not feasible to capture an ideal clean image from the wild. Rather, a long sequence of aligned noisy images must be captured beforehand [5] to synthesize the pseudo-clean reference. Thus, only a few clean images are available from the same scene distribution of the noisy images in a real situation,

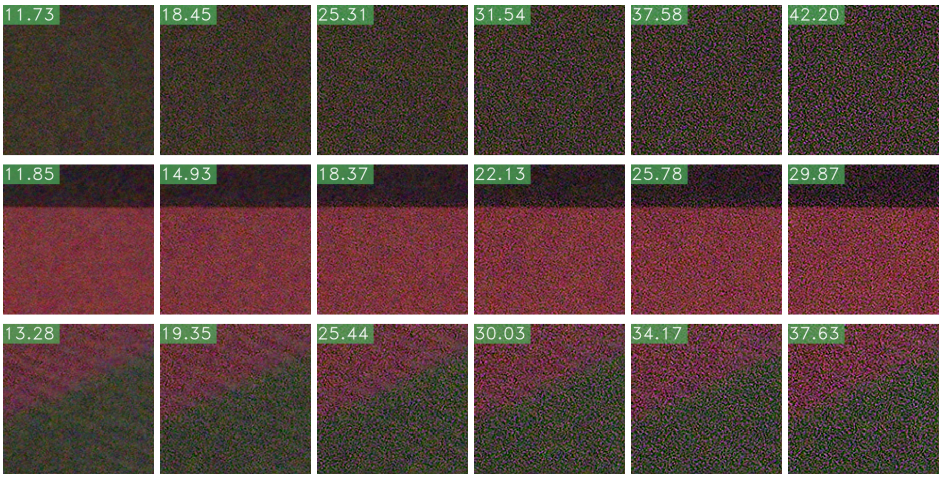
which can be considered as the first data constraint we handle here. Secondly, once the noise generator is trained on the desired distributions P_N and P_C, it should be able to produce pseudo-noisy images paired to any clean image $x \sim P'_C$ to train a denoising model. Figure 4.6 shows a variety of scenes in each different datasets. They differ in many points, such as types and scale of the contents, brightness, and color temperature.

The existing generative noise modeling methods [22, 23, 25] used large amount of samples in P_C, and assumed P'_C to be same as P_C. Such a setting is possible only if a sufficiently large noisy and clean image dataset is given, which is not a practical situation. Our method overcomes the limitation of this problem that have not been explored before and can be used in general real situations. To verify this, we use Urban100 [8] along with the other datasets mentioned above as the samples in P'_C.

Number of Samples $\sim P_C$	P'_C	PSNR(dB)	SSIM
36K (100%)	S	34.08	0.909
	D	31.72	0.826
	B	31.74	0.825
	U	31.32	0.803
18K (50%)	S	33.53	0.882
	D	30.68	0.760
	B	29.96	0.742
	U	29.72	0.741
720 (2%)	S	31.98	0.847
	D	29.36	0.745
	B	29.27	0.735
	U	29.21	0.738
360 (1%)	S	31.84	0.849
	D	29.35	0.740
	B	29.08	0.733
	U	29.24	0.739

Table 4.4: Denoising performance of our C2N under data constraints. S, D, B, U denote the SIDD, the DIV2K high-resolution images, the BSD traning images, and the Urban100 dataset, respectively. P_C is fixed to S for all experiments. Evaluation is done on the SIDD validation set.

Table 4.4 shows that our method preserves its performance when there are two data constraints, (1) where not enough clean images in P_C are given to train the noise generator, (2) where the clean images from different scene distribution P'_C are used to train a denoising model. Our method already uses surprisingly small amount of samples for training the C2N model, compared to $\sim$500K image patches of 64×64 size used in the previous generative noise modeling methods [23, 25]. The C2N model trained with much smaller amount of clean images in P_C still shows performance comparable to previous unsupervised denoising methods. Also, for the case of P'_C to be different to P_C, our C2N can still train the following denoising model with its generated pseudo-noisy data.



(a) $\lambda = 0.0$ (b) $\lambda = 0.2$ (c) $\lambda = 0.4$ (d) $\lambda = 0.6$ (e) $\lambda = 0.8$ (f) $\lambda = 1.0$

Figure 4.7: Image generation with interpolated r. For the same clean image patch, we interpolate two different r vectors with a factor λ and obtain the resulting images. Each column is generated using same vector r. The standard deviation of each residual noise map is displayed in each image. Best with zoomed.

4.5 Generating noise by interpolation in latent space

We also studied the effects of r, the random vector input of the generator. in Figure **??**, we sampled two r vectors that correspond to different pseudo-noisy images and visualize the generated images with interpolated r as following equation, $r = (1-\lambda)r_1+\lambda r_2$. Here, r_1 and r_2 are the two sampled vectors and λ is interpolation factor between the two vectors. The qualitative results show that our C2N can generate real-world noise that corresponds to varying conditions, such as strong or weak noise from various camera types in the SIDD [5].

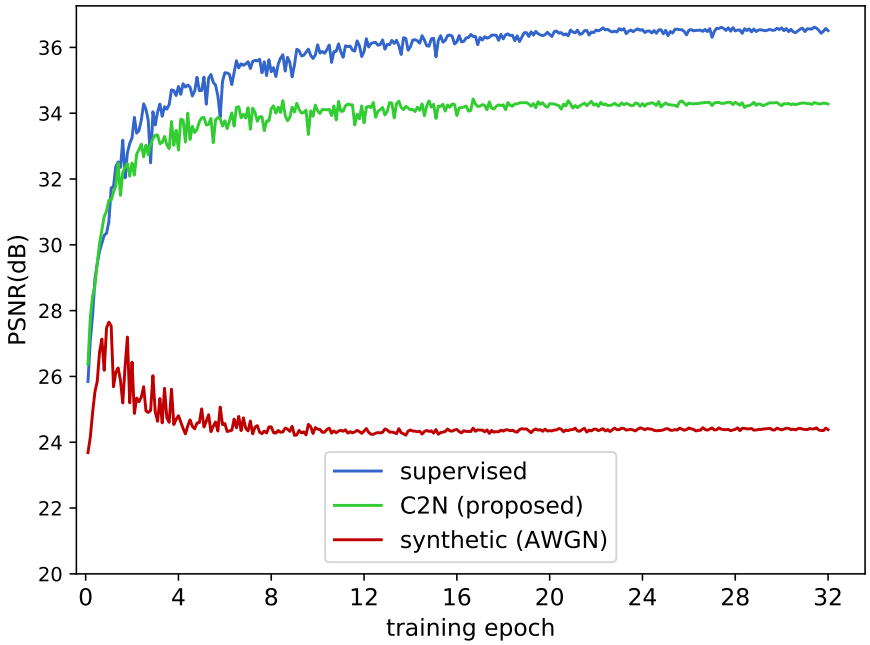


Figure 4.8: Denoising performance on SIDD validation set according to training epochs. The blue, green, and red learning curves correspond to those of the DnCNN [9] trained to denoise true noisy images in SIDD training set, pseudo-noisy images generated by our C2N, and noisy images synthesized by AWGN, respectively.

4.6 Verifying C2N in Denoiser Training

To show that the distribution of generated noise is close enough to that of the ground truth, we observe the difference in the learning curves of denoising models according to their training noisy images. Suppose a denoising model learns to remove non-feasible noise. In that case, although it helps to restore the clean image for some time in the first place, the model eventually overfits to removing the wrong noise and cannot perform correct denoising.

Figure 4.8 shows that this phenomenon does not appear while learning to denoise using C2N, unlike learning in an unrealistic synthetic noise. For AWGN synthetic

noise, we randomly sampled the standard deviation from $[0.24, 11.51]$, a range that fits the noise of the SIDD training dataset, following [23].

Chapter 5

Conclusion

In this paper, we propose a framework for practical real-world denoising which includes our novel noise generator and a denoising model. By explicitly designing each components of the generator considering characteristics of general real-world noise, it can successfully learn to simulate the noise distribution of the noisy images, without using any paired clean images. Using the generated data, we train an arbitrary denoising model to outperform the existing denoising methods without use of actual data pairs. Furthermore, our method preserves its higher performance when only small amount of ground truth images are available and they are not approachable during training of a denoising model. We believe our method can be a key to solve the challenging points of practical real-world denoising.

Bibliography

[1] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *CVPRW*, 2019.

[2] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

[3] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[4] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.

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

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

[7] 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*. IEEE, 2001.

[8] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015.

[9] 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.

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

[11] 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.

[12] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[13] 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.

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

[15] Chang Chen, Zhiwei Xiong, Xinmei Tian, and Feng Wu. Deep boosting for image denoising. In *ECCV*, 2018.

[16] Yang Liu, Saeed Anwar, Liang Zheng, and Qi Tian. Gradnet image denoising. In *CVPRW*, 2020.

[17] Sungmin Cha and Taesup Moon. Gan2gan: Generative noise learning for blind image denoising with single noisy images. *arXiv preprint arXiv:1905.10488*, 2019.

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

[19] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

[20] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. *arXiv preprint arXiv:1901.11365*, 2019.

[21] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.

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

[23] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019.

[24] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *NIPS*, 2014.

[25] Zhiwei Hong, Xiaocheng Fan, Tao Jiang, and Jianxing Feng. End-to-end unpaired image denoising with conditional adversarial networks. In *AAAI*, 2020.

[26] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *CVPR*, 2019.

[27] Josue Anaya and Adrian Barbu. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018.

[28] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018.

[29] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *CVPR*, 2020.

[30] Dong-Wook Kim, Jae Ryun Chung, and Seung-Won Jung. Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling. In *CVPRW*, 2019.

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

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

[33] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *CVPRW*, 2018.

[34] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, 2017.

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

[36] Zhihao Xia and Ayan Chakrabarti. Identifying recurring patterns with deep neural networks for natural image denoising. In *WACV*, 2020.

[37] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[38] Ke Yu, Xintao Wang, Chao Dong, Xiaoou Tang, and Chen Change Loy. Path-restore: Learning network path selection for image restoration. *arXiv preprint arXiv:1904.10343*, 2019.

[39] Coleman Broaddus, Alexander Krull, Martin Weigert, Uwe Schmidt, and Gene Myers. Removing structured noise with self-supervised blind-spot networks. In *ISBI*, 2020.

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

[41] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*, 2020.

[42] Huangxing Lin, Weihong Zeng, Xinghao Ding, Xueyang Fu, Yue Huang, and John Paisley. Noise2blur: Online noise extraction and denoising. *arXiv preprint arXiv:1912.01158*, 2019.

[43] Rui Zhao, Daniel PK Lun, and Kin-Man Lam. Ntgan: Learning blind image denoising without clean reference. *arXiv preprint arXiv:2009.04286*, 2020.

[44] Gerald C Holst. Ccd arrays, cameras, and displays. 1998.

[45] Alan C Bovik. *The essential guide to image processing*. Academic Press, 2009.

[46] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[47] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[48] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[51] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1865–1873, 2016.

[52] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[53] Samuel Hurault, Thibaud Ehret, and Pablo Arias. Epll: an image denoising method using a gaussian mixture model learned on a large set of patches. *Image Processing On Line*, 8:465–489, 2018.

[54] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *TPAMI*, 39(6):1256–1272, 2016.

초 록

학습 기반 영상 잡음 제거 모델의 사용은, 잡음이 있는 이미지들과 깨끗한 이미지들이 잘 정렬된 쌍을 이룬 상태로 제공되거나, 주어진 잡음의 분포로부터 학습용 샘플들을 합성할 수 있는 상황에 한정되어 있다. 최근의 생성모델 기반의 방법들은 실제 잡음의 분포가 알려지지 않은 경우에도 그것을 정확하게 시뮬레이션하는 방법론을 도입하고 있지만, 몇 가지 제한점들이 여전히 존재한다. 기존의 그러한 방법들은 실제 잡음의 분포를 얻을 수 있는 데이터가 주어지거나 잡음에 대해 비현실적인 가정이 내려진 경우로 적용 범위가 제한되었다. 실제 상황에서의 잡음 생성모델은 잡음이 있는 이미지와 깨끗한 이미지의 쌍을 사용하지 않고도 복잡하며 일반적인 잡음의 시뮬레이션을 학습할 수 있어야 한다. 이러한 실제적 상황에서 학습한 잡음 생성모델은 복잡한 잡음의 분포가 아닌 특정 질감의 패턴을 만들어내는 동작을 하게 되어버리기 쉽기에, 이 문제를 해결하기 위해 설계한 모델 구조를 제안한다. 이렇게 설계한, C2N 즉 Clean-to-Noisy 영상 생성 프레임워크를 개발하여 복잡한 실영상의 잡음을 어떠한 쌍을 이룬 학습 데이터 없이 모방할 수 있다. 이 C2N을 기존의 잡음 제거 모델과 결합하는 것으로 실영상 잡음 제거 벤치마크에서 기존의 비감독 학습 방법들을 큰 폭으로 능가할 수 있으며, 이를 통해 제안 방법의 효과를 검증한다. 또한 이전의 잡음 생성모델 방법들에 의해선 탐구되지 않았던 영역인, 데이터에 대한 여러 제약이 있는 실용적 상황에 대해 본 방법을 확장한다.

주요어: 영상 잡음 제거, 영상복원, 실영상 잡음 제거, 생성모델, 적대적 생성 신경망, 비감독 잡음 제거

학번: 2018-27051

ACKNOWLEGEMENT

졸업에 이르기까지 많은 분들로부터 직간접적으로 많은 영향과 도움을 받았습니다. 언제나 제게 과분할 정도의 포용과 열정으로 지도해주신 이경무 지도교수님께, 많은 부분에 있어서 신경 써 주신 연구실의 여러 멋진 분들께, 그리고 졸업 논문 심사를 해 주신 조남익 교수님과 한보형 교수님께 감사 말씀을 드립니다. 또한 이러한 하나의 마무리 및 앞으로에 대한 선택을 만들어 가는 과정에서 항상 마음 둘 곳이 되어주신 모든 분들께 감사드립니다.