Perception-based multiplicative noise removal with Diffusion models

An Vuong Thinh Nguyen

Electrical Engineering & Computer Science
College of Engineering, Oregon State University

Introduction

Recently, deep learning based methods have been introduced with great successes in denoising performance [1][2][3][4]. These methods usually use image-to-image translation architecture, where the neural networks directly predict the clean images, or the amount of noise generated by the stochastic process, without much assumption on the noise dynamics. Thus, many of these models can be applied to reverse different kinds of corruptions, including multiplicative noise. However, these techniques mostly rely on "per-pixel" metrics such as MSE, PSNR, or SSIM, which has been observed to not correlate well with human perception [5].

In this work, we propose the novel application of **Stochastic Differential Equations (SDEs) to perform multiplicative noise removal**. We show that the dynamics of multiplicative noise is well captured by SDEs, specifically *Geometric Brownian motion*. We then derive the reverse SDEs which are used to generate denoised samples. By running extensive experiments on two different datasets, we demonstrate the effectiveness of our method on creating clean images that achieve high perception scores.

Experiments

We ran our experiments on CelebA and UC Merced Land Use datasets, using U-Net as the backbone architecture for our neural networks. The training was done on 100,000 images from the CelebA dataset, while testing was performed on 2,096 images from CelebA holdout set and another 2,096 images from UC Merced Land Use, images were resized to 224x224 pixels. We did not finetune on the land use dataset since we wanted to test the generalization of directly modeling the noise dynamics. We made comparisons with a wide range of methods, from classical techniques to modern neural denoisers. A sample is provided in Figure 1.

Dataset		CelebA			
Noise level	Model	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
0.12	Ours (ODE)	**24.3077**	**0.0774**	28.5567	0.8994
	DeblurGAN	38.3176	0.0902	24.3920	0.8360
	Restormer	29.6475	0.0848	**31.5912**	**0.9473**
	MPRNet	31.5731	0.0906	31.4240	0.9447
	NAFNet	27.0304	0.0805	31.5588	0.9466
	DnCNN	33.2722	0.1113	30.5007	0.9281
	SRAD	70.4303	0.3853	24.3354	0.7311
	CBM3D	36.2575	0.2128	26.1749	0.8481

Table 1. Comparison of different methods on CelebA dataset at noise level 0.12

Dataset		UC Merced LandUse			
Noise level	Model	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
0.12	Ours (ODE)	**63.2679**	**0.1333**	28.8183	0.8727
	DeblurGAN	93.9357	0.1916	26.1810	0.8218
	Restormer	76.1931	0.1653	**30.2727**	**0.9016**
	MPRNet	98.2022	0.1884	30.0573	0.8965
	NAFNet	77.2864	0.1644	30.0889	0.8977
	DnCNN	138.2030	0.2361	29.2134	0.8710
	SRAD	107.6315	0.4712	25.2329	0.7247
	CBM3D	100.1406	0.3570	27.1042	0.8144

Table 2. Comparison of different methods on UC Merced LandUse dataset at noise level 0.12

Methods

Given a real-valued signal $\mathbf{x} \in \mathbb{R}^d$, with the assumption that corruption $\boldsymbol{\epsilon} \in \mathbb{R}^d$ affects each component independently. Denoting $\tilde{\mathbf{x}}$ as the corrupted version

$$\tilde{\mathbf{x}} = \mathbf{x} \odot \boldsymbol{\epsilon} \tag{1}$$

Then (1) can be well modeled by the following SDE

$$d\mathbf{x} = \alpha(t)\mathbf{x}(t) \odot d\boldsymbol{\beta}(t) \tag{2}$$

where $\alpha(t)$ is some time-varying scalar function and $\boldsymbol{\beta}(t)$ is a Brownian motion on $\mathbb{R}^d$. Indeed, the solution to (2) is given as

$$x_{t,i} = x_{0,i} \exp\Big(-\int_0^t \frac{1}{2}\alpha^2(\tau)d\tau + \Big(\int_0^t \alpha^2(\tau)d\tau\Big)^{\frac{1}{2}} n\Big) \tag{3}$$

where $x_{t,i}$ denotes the i-th entry of $\mathbf{x}(t)$, and $n \sim \mathcal{N}(0,1)$. Since n is Gaussian, the exponential term in (3) follows Log-normal distribution, satisfying our previous assumption on ϵ. If we select $\mathbf{x}(0)$ to be the clean image $\mathbf{x}$, then with appropriate value of t, $x(t) = \tilde{\mathbf{x}}$ is well modeled by (2).

Let us denote $y_{t,i} = \log x_{t,i}$. Now, equation (3) becomes

$$y_{t,i} = y_{0,i} - \int_0^t \frac{1}{2}\alpha^2(\tau)d\tau + \int_0^t \alpha(\tau)d\beta(\tau) \tag{4}$$

which can also be expressed in differential vector form to obtain the SDE

$$d\mathbf{y}_t = -\frac{1}{2}\alpha^2(t)\mathbf{1}dt + \alpha(t)d\boldsymbol{\beta}(t) \tag{5}$$

This has the corresponding time-reversal

$$d\mathbf{y}_{T-t} = \Big(\frac{1}{2}\alpha^2(T-t)\mathbf{1} + \alpha^2(T-t)\nabla \log p_{T-t}(\mathbf{y}_{T-t})\Big)dt + \alpha(T-t)d\boldsymbol{\beta}(T-t) \tag{6}$$

where T is the terminal time, i.e. at which the forward SDE (5) stopped. Applying Euler-Maruyama discretization to (5) and (6), where $\alpha(t)$ is selected to be $\sqrt{\frac{d\sigma(t)}{dt}}$ with $\sigma(t)$ is some differentiable function having non-negative slope, gives the following pair of SDEs

$$\mathbf{y}_k = \mathbf{y}_0 - \frac{1}{2}\big(\sigma(k) - \sigma(0)\big)\mathbf{1} + \sqrt{\sigma(k) - \sigma(0)}\mathbf{n}_k \tag{7}$$

$$\mathbf{y}_{K-k} = \mathbf{y}_{K-k+1} + \frac{1}{2}\Big(\sigma(K-k+1) - \sigma(K-k)\Big)\Big(\mathbf{1} + 2\nabla \log p_{K-k+1}(\mathbf{y}_{K-k+1})\Big) + \sqrt{\sigma(K-k+1) - \sigma(K-k)}\mathbf{n}_k, \qquad \mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \tag{8}$$

From (7) and (8), let $\mathbf{s}_{\boldsymbol{\theta}}$ be a neural network that learns $\nabla \log p_{K-k+1}(\mathbf{y}_{K-k+1})$, it can be shown that the desired training objective is

$$\mathcal{L}_{\boldsymbol{\theta},\text{discete}} = \mathbb{E}_{\mathbf{y}}\mathbb{E}_k\Big[||\boldsymbol{n}_k + \sqrt{\sigma(k) - \sigma(0)}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{y}_k, k)||_2^2\Big] \tag{9}$$

After training, sampling can be carried out by (7), ODE sampler [6], or DDIM sampler [7]. Ablation of these samplers is shown in Table 3.

Sampling technique	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
ODE	**13.9156**	**0.0365**	**31.8902**	**0.9348**
DDIM	25.3188	0.0882	28.6549	0.9032
Stochastic	32.3811	0.1075	26.8267	0.8493

Table 3. Comparison of different sampling techniques on CelebA dataset at noise level 0.12

Denoising samples

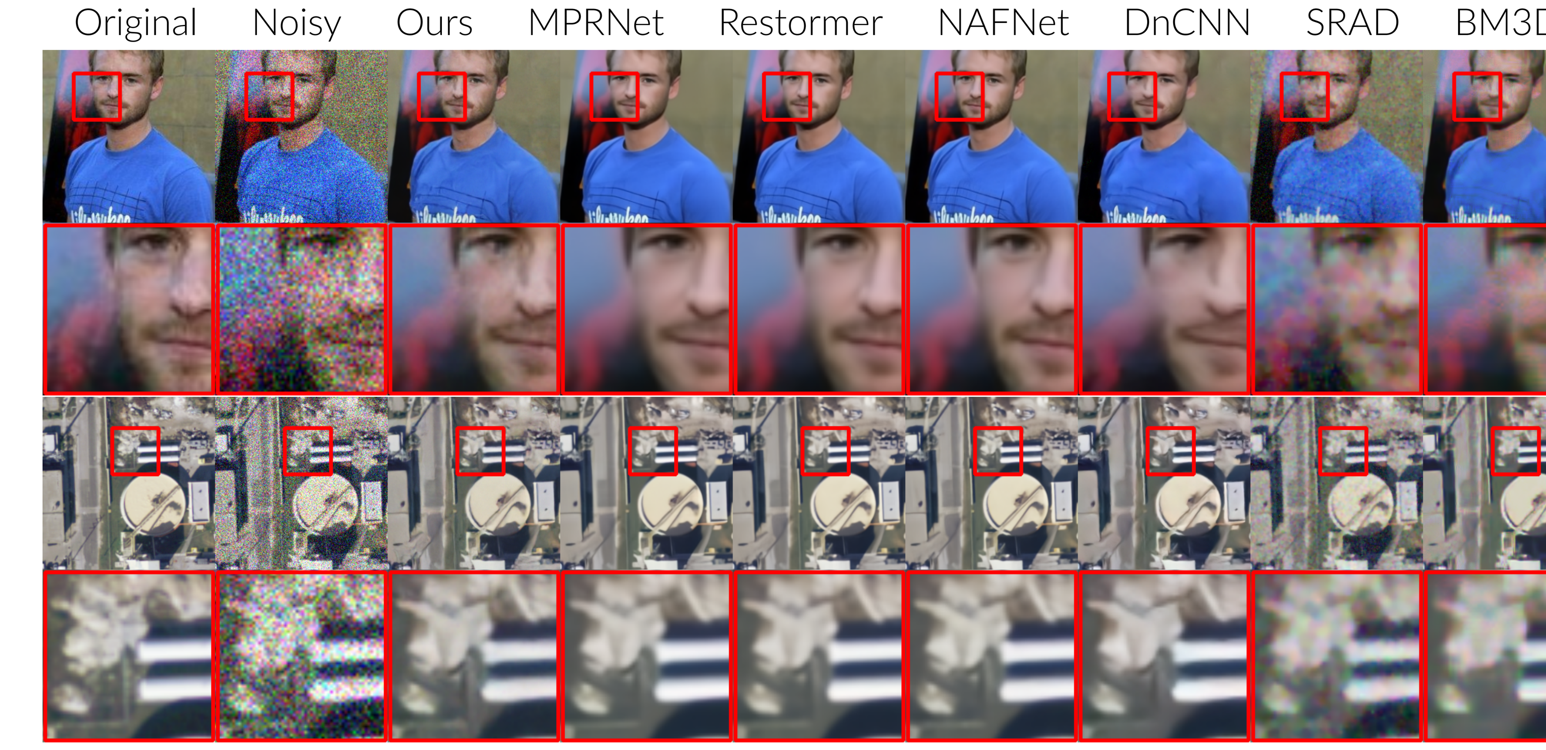

Figure 1. Different denoising models on randomly selected CelebA and LandUse images, at noise level 0.12.

Conclusion

In conclusion, we introduced a novel SDE-based diffusion model for removing multiplicative noise. The work presented the construction of the forward and reverse SDEs that directly captures the dynamics of the noise model. In addition, it also established the training objective as well as multiple different sampling equations based on Probability flows and DDIM techniques. The proposed model was compared to classical image processing algorithms, including BM3D and SRAD, as well as the modern CNN-based methods, demonstrating that our method outperforms the current state-of-the-art denoising models in perception-based metrics across all noise levels, while still remaining competitive in PSNR and SSIM.

References

[1] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[2] D. Li, W. Yu, K. Wang, D. Jiang, and Q. Jin, "Speckle noise removal based on structural convolutional neural networks with feature fusion for medical image," *Signal Processing: Image Communication*, vol. 99, p. 116500, 2021.

[3] H. Choi and J. Jeong, "Speckle noise reduction in ultrasound images using srad and guided filter," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–4, IEEE, 2018.

[4] D. Feng, W. Wu, H. Li, and Q. Li, "Speckle noise removal in ultrasound images using a deep convolutional neural network and a specially designed loss function," in *Multiscale Multimodal Medical Imaging: First International Workshop, MMMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1*, pp. 85–92, Springer, 2020.

[5] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6228–6237, 2018.

[6] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[7] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.