Perception-based multiplicative noise removal with Diffusion models

An Vuong
Electrical Engineering & Computer Science
Oregon State University
Corvallis, USA
vuonga2@oregonstate.edu

Thinh Nguyen
Electrical Engineering & Computer Science
Oregon State University
Corvallis, USA
thinhq@eecs.oregonstate.edu

***Abstract*—We present a novel approach to perform multiplicative noise removal, utilizing the recent developments of diffusion models. We show that multiplicative noise, which commonly appears in images produced by synthetic aperture radar (SAR), laser, or optical lenses, can be well-modeled by a Geometric Brownian process in the logarithmic domain. This process admits a time-reversal stochastic differential equation (SDE), which is utilized to perform noise removal. We conduct extensive experiments to compare our approach with classical methods as well as state-of-the-art Deep Learning-based approaches. Our models significantly outperform others in terms of perception-based metrics such as LPIPS and FID, while remaining competitive in traditional pixel-based metrics like PSNR and SSIM.**

***Index Terms*—multiplicative noise, diffusion process, brownian motion, stochastic differential equations**

I. Introduction

Multiplicative noise removal is a long standing problem in computer vision and has been studied by many researchers over the past few decades. Unlike additive noise, which is usually the result of thermal fluctuations during image acquisition or transmission, multiplicative noise happens when multiple copies of the signal with random scaling factors are added together. This often happens due to the internal physical construction of the image capturing devices, i.e. optical lenses, radar/laser imaging, ultrasound sensors, etc. Because of this, removing multiplicative noise, sometimes referred to as despeckling, often requires more sophisticated approaches compared to its counterpart additive noise. Popular approaches include modelling the noise using Partial Differential Equations (PDEs) [1] , converting into additive domain and optimize using Total Variation (TV) objective [2], and applying MAP estimation [3]. Classical methods based on block-matching technique also works well for this problem [4].

Recently, deep learning based methods have been introduced with great successes in denoising performance [5] [6] [7] [8]. 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 [9].

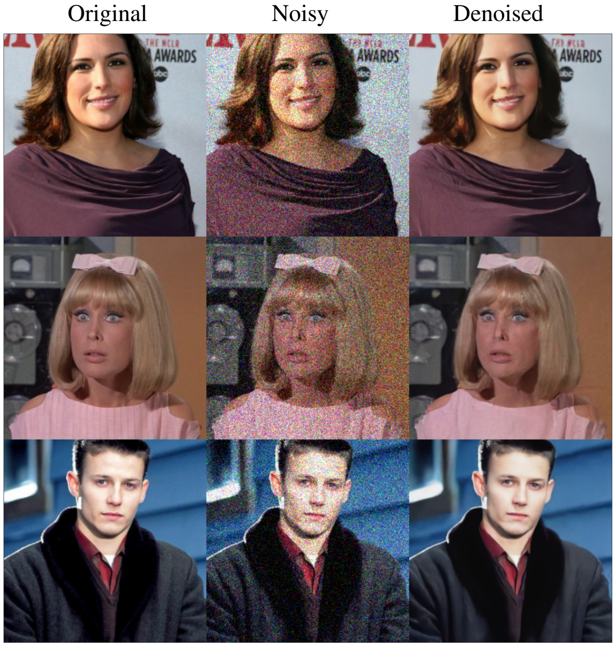

Fig. 1. Samples generated by our methods, on images randomly selected from CelebA dataset. From left to right are the original, corrupted by multiplicative noise (noise level 0.08), and denoised versions.

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. We discuss the construction of our approach in Section III, with experiments details in Section IV.

II. Related work

Over the last few years, with the advances of deep learning, there has been active research in applying convolutional neural network (CNN) to image denoising problems, especially for speckle or multiplicative noise. Notable works can be found in [6] [7] [8]. In these texts, the common theme is to perform image-to-image translation with a CNN acting as the mapping function. This CNN is usually trained to minimize MSE or PSNR loss directly on the pairs of clean and noisy images. Some works propose to use specially-crafted features as the input, such as frequency features [10], or wavelet features [11], and sub-bands [12]. These works are usually limited to grayscale images, and are often matched in performance by DnCNN [5], and outperformed by NAFNet [13], MPRNet [14], or Restormer [15].

Recently, there is a line of works applying diffusion technique to this problem. In [16], the authors propose a DDPM-like architecture for despeckling, but this model needs to be re-trained for each noise level. Similarly, [17] [18] also use DDPM framework with minor modifications. These works still limit their testing to greyscale images only. We find the discussion in [19] to be the most related to our work, albeit with different assumption of the noise characteristics and the construction of the diffusion process, where the authors still rely on the DDPM equations.

To the best of our knowledge, we are the first to directly model this problem using SDE, which captures the dynamics of the noise process, and derive the sampling equation which is then used to perform denoising.

III. Methods

A. Diffusion models

This section gives a brief overview of SDEs, Itô's calculus, and the application to generative modeling.

Let $\beta(t)$ be a Brownian motion indexed by time t, i.e. $\beta(t)$ is a random process with independent and zero-mean Gaussian increment, then the classic result from Itô's calculus gives

$$\frac{1}{2}d\beta^2(t) = \beta(t)d\beta(t) + \frac{1}{2}dt \tag{1}$$

Utilizing this result, one can solve the following SDE

$$dx(t) = f(x(t), t)dt + L(x(t), t)d\beta(t) \tag{2}$$

where $f(.)$, $L(.)$ are some functions, $x(t)$ is the random process of interest, and $dx(t)$ represents the (random) infinitesimal change of $x(t)$. Notably, under certain smoothness assumptions, there exists a unique SDE that models the reverse process:

$$\begin{aligned} dx(T-t) = &- f(x(T-t), T-t)dt \\ &+ L(x(T-t), T-t)d\beta(T-t) \\ &+ L^2(T-t)\nabla \log p_{T-t} dt \end{aligned} \tag{3}$$

where p_{T-t} denotes the distribution of $x(T-t)$. In-depth discussions regarding the proof and existence of this process can be found in [20].

If $\nabla \log p_{T-t}$ is known, one can can run (3) to generate new sample $x(T-t)$ that comes from data distribution $p_{T-t}(x)$. This motivates the search for an efficient method to estimate $\nabla \log p_{T-t}$, also known as the score function. Let $s_{\boldsymbol{\theta}}(x)$, parameterized by $\boldsymbol{\theta}$, be the estimation of $\nabla \log p(x)$. Then [21] provides an efficient way to compute $s_{\boldsymbol{\theta}}(x)$ without needing access to $\nabla \log p(x)$. This result was used in [22] to train neural networks for images generation tasks that achieve high-quality results, with a small tweak, the model learns $\nabla \log p_{T-t}(x)$, instead of $\nabla \log p(x)$, leading to more stable training. This approach is referred to as Score-based Generative Models (SGMs).

In a parallel development, [23] proposed a similar framework from Markov chain perspective, named Diffusion Denoising Probabilistic Models (DDPM). DDPM formulation yields a simple regression objective, which is shown to be equivalent to SGM's objective under Gaussian noise assumption [24]. In this paper, we use the SDE-based formulation because of its flexibility, allowing us to directly model the desired underlying dynamics of the noise process.

B. Noise models

In this section, we introduce multiplicative noise, and show how SDE can be used to model the dynamics of this process.

A real-valued signal $x \in \mathbb{R}$ is corrupted by multiplicative noise is modeled as

$$\tilde{x} = \epsilon x \tag{4}$$

where $\epsilon \in \mathbb{R}$ is a random variable, usually modeled as having Gamma or Log-normal distribution [25], and $\tilde{x}$ is the corrupted version of x. Here, we extend this noise process to multi-dimensional $\mathbf{x} \in \mathbb{R}^d$, with the assumption that this corruption affects each component independently

$$\tilde{\mathbf{x}} = \mathbf{x} \odot \boldsymbol{\epsilon} \tag{5}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^d$ and $\odot$ represents the element-wise multiplication. We now show that (5) can be well modeled by the following SDE

$$d\mathbf{x} = \alpha(t)\mathbf{x}(t) \odot d\boldsymbol{\beta}(t) \tag{6}$$

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 (6) (proofs omitted due to space constraints) is given as

$$x_{t,i} = x_{0,i} \exp\Big(- \int_0^t \frac{1}{2}\alpha^2(\tau)d\tau + \Big(\int_0^t \alpha^2(\tau)d\tau \Big)^{\frac{1}{2}} n \Big) \tag{7}$$

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 (7) 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 (6).

We can now apply Anderson's theorem to derive the reverse SDE for (6) and use score-matching technique to construct a denoising model. But this formulation gives a rather complicated reverse SDE. Instead, we propose to apply a simple logarithmic transformation to $\mathbf{x}$, this yields a much simpler reverse SDE, with the additional advantage of being able to apply the results from [24], making the loss function easier to derive.

C. Loss function in the logarithmic domain

Let us denote $y_{t,i} = \log x_{t,i}$. Now, equation (7) becomes

$$y_{t,i} = y_{0,i} - \int_0^t \frac{1}{2}\alpha^2(\tau)d\tau + \int_0^t \alpha(\tau)d\beta(\tau) \tag{8}$$

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

$$d\mathbf{y}_t = -\frac{1}{2}\alpha^2(t)\mathbf{1}dt + \alpha(t)d\boldsymbol{\beta}(t) \tag{9}$$

This has the corresponding time-reversal (proofs omitted due to space constraints)

$$\begin{aligned} d\mathbf{y}_{T-t} = \Big(\frac{1}{2}\alpha^2(T-t)\mathbf{1} + \alpha^2(T-t)\nabla \log p_{T-t}(\mathbf{y}_{T-t})\Big)dt \\ + \alpha(T-t)d\boldsymbol{\beta}(T-t) \end{aligned} \tag{10}$$

where T is the terminal time index, i.e. at which the forward SDE (9) stopped.

Applying Euler-Maruyama discretization to (9) and (10), 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

$$\begin{aligned} \mathbf{y}_k &= \mathbf{y}_{k-1} - \frac{1}{2}\big(\sigma(k) - \sigma(k-1)\big)\mathbf{1} \\ &\quad + \sqrt{\sigma(k) - \sigma(k-1)}\mathbf{n}_k \\ &= \mathbf{y}_0 - \frac{1}{2}\big(\sigma(k) - \sigma(0)\big)\mathbf{1} + \sqrt{\sigma(k) - \sigma(0)}\mathbf{n}_k \end{aligned} \tag{11}$$

$$\begin{aligned} \mathbf{y}_{K-k} &= \mathbf{y}_{K-k+1} \\ &\quad + \frac{1}{2}\Big(\sigma(K-k+1) - \sigma(K-k)\Big)\Big(\mathbf{1} + \\ &\qquad 2\nabla \log p_{K-k+1}(\mathbf{y}_{K-k+1})\Big) \\ &\quad + \sqrt{\sigma(K-k+1) - \sigma(K-k)}\mathbf{n}_k \\ &\quad \mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \end{aligned} \tag{12}$$

To derive the loss function, note that equation (11) has a Gaussian transition kernel $p(\mathbf{y}_k|\mathbf{y}_{k-1}) = \mathcal{N}\big(\mathbf{y}_k; \mathbf{y}_{k-1} - \frac{1}{2}\big(\sigma(k) - \sigma(k-1)\big)\mathbf{1}, \sigma(k) - \sigma(k-1)\big)$. Thus, results from [24] applies, which states the following connection between SGMs and DDPMs: let $\mathbf{s}^*(\mathbf{x}_t, t)$ and $\mathbf{n}^*(\mathbf{x}_t, t)$ be the minimizers of SGM and DDPM objectives, respectively, then $\mathbf{n}^*(\mathbf{x}_t, t) = -\sqrt{\text{var}(\mathbf{x}_t)}\mathbf{s}^*(\mathbf{x}_t, t)$ if $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is Gaussian, where $\text{var}(\mathbf{x}_t)$ denotes the variance of the stochastic process at time t. This means the denoising objective from DDPM can be readily applied to our formulation in the logarithmic domain, giving the following trainable loss

$$\mathcal{L}_{\boldsymbol{\theta},\text{discete}} = \mathbb{E}_{\mathbf{y}}\mathbb{E}_k\Big[||\boldsymbol{n}_k + \sqrt{\sigma(k) - \sigma(0)}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{y}_k, k)||_2^2\Big] \tag{13}$$

D. Sampling techniques

In practice, instead of sampling directly using (12), Probability flows ODE [26] or Implicit probabilistic models (DDIM) [22] are often used to enhance image quality and sampling efficiency (i.e. reducing number of time steps). By experimenting, we find that Probability flows ODE approach produces the best result for our purpose. The ODE sampling equation is given as (proofs omitted due to space constraints)

$$\begin{aligned} \mathbf{y}_{K-k} = \mathbf{y}_{K-k+1} + \frac{1}{2}\Big(\sigma(K-k+1) - \sigma(K-k)\Big)\Big(\mathbf{1} + \\ \nabla \log p_{K-k+1}(\mathbf{y}_{K-k+1})\Big) \end{aligned} \tag{14}$$

For completeness, we also provide the corresponding DDIM sampling equations (15), as well as quantitative comparisons between ODE, DDIM, and the vanilla Stochastic samplers in Table I. Some qualitative examples are provided in Fig. 2.

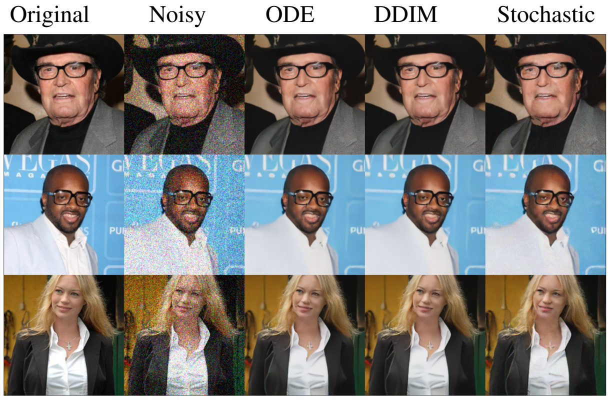

Fig. 2. Comparing between different sampling techniques on randomly selected CelebA images, at noise level 0.12. The first two columns include the original images and their noised versions, respectively. These are followed by the results generated by our model using ODE, DDIM, and stochastic samplers, respectively.

$$\begin{aligned} \mathbf{y}_{k-1} &= \hat{\mathbf{y}}_0 - \frac{1}{2}\eta(k-1)\mathbf{1} \\ &\quad + \frac{\sqrt{\eta(k-1)}}{\sqrt{\eta(k)}}\big(\mathbf{y}_k - \hat{\mathbf{y}}_0 + \frac{1}{2}\eta(k)\mathbf{1}\big) \\ \hat{\mathbf{y}}_0 &= \mathbf{y}_k + \frac{1}{2}\eta(k)\mathbf{1} + \eta(k)\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{y}_k, k) \end{aligned} \tag{15}$$

Thus, by transforming the data into the logarithmic domain, we can take advantage of the simplicity of DDPM framework, while still being able to accurately capture our noise model thanks to the flexibility of SDE. Furthermore, since $\log$ is a bijective transformation for positive variables, we can easily recover our images from the samples generated in logarithmic domain by taking the exponential.

The appendix and Pytorch implementation are publicly available at:
https://github.com/anvuongb/sde_multiplicative_noise_removal.

IV. Experiments

A. Experiment settings

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.

The diffusion model is trained by optimizing (13), then the noise removal process is performed using (14) to sample from the trained model. For baseline models, we have selected several denoising methods, ranging from classical signal processing techniques to modern CNN-based frameworks:

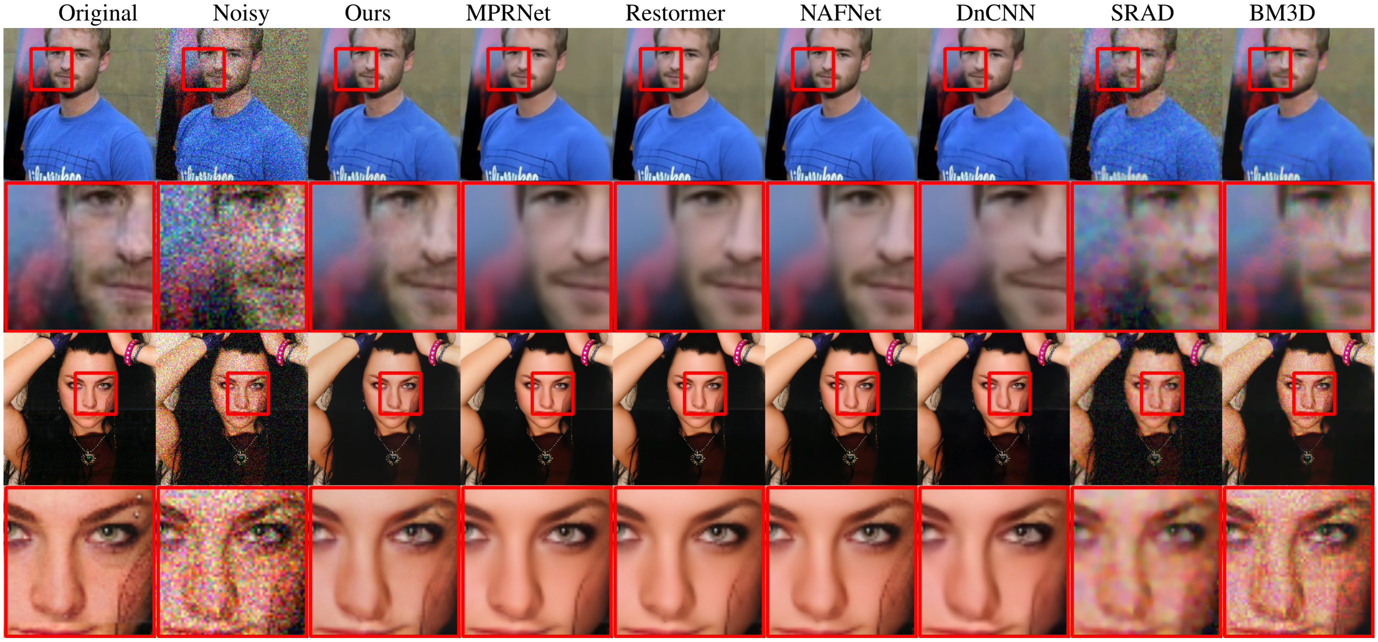

Fig. 3. Comparing between different denoising models on randomly selected CelebA image, at noise level 0.12.

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 I
COMPARISON OF DIFFERENT SAMPLING TECHNIQUES ON CELEBA DATASET AT NOISE LEVEL 0.12

- Block-matching and 3D filtering (BM3D) [4].
- Speckle reducing anisotropic diffusion (SRAD) [1].
- Gaussian denoising in latent space DnCNN [5] .
- Transfomer-based: MPRNet [14] and Restormer [15].
- NAFNet [13], the current state-of-the-art in noise removal and image restoration.
- GAN-based DeblurGAN [27].

Regarding training settings, we trained our model with $T = 500$ diffusion steps, linear noise schedule $\sigma(k) \in [0.0001, 0.02]$, and Adam optimizer. For DnCNN, NAFNet, MPRNet, Restormer, and DeblurGAN, we followed the optimal training options provided by the authors. All models are trained for 100 epochs. Training was done on 2x RTX3090 under Ubuntu using Pytorch framework. Since DnCNN, NAFNet, MPRNet, Restormer, and DeblurGAN need to be trained for some specific noise levels, we chose two different noise variances $[0.04, 0.12]$ for the noise term in (7), this corresponds to $T = 100, 300$ in our diffusion process formulation. We note that, while these models need to be trained for each noise level, our framework only needs to be trained once, then inference can be run at different noise levels since the noise dynamics is already captured by the SDE formulation.

For evaluation metrics, we chose to evaluate on perception-based metrics FID, LPIPS and traditional signal processing metrics PSNR, SSIM. While PSNR and SSIM are important measurements in computer vision, they have been shown to not correlate well with human perception on image restoration tasks [28]. In our experiments, we also observed substantial discrepancies between these two types of metrics, more detailed discussion is provided in the next section.

B. Experiment results

Results on CelebA dataset.[1] Quantitative results are presented in the left half of Table II. Compared to the state-of-the-art models NAFNet and Restormer, our method shows slightly worse performance in per-pixel metrics, while achieving significantly better FID and LPIPS scores, across all tested noise levels. This is also true when comparing to DnCNN and MPRNet. We note that these models were architected to directly optimize for PSNR during training, thus they strive to achieve the best fidelity at the cost of diverging from the input distribution. For classical methods, CBM3D (the RGB version of BM3D) performs respectably, sometimes even coming close to DnCNN. In contrast, SRAD falls far behind in all metrics, we suspect this is because the tests were conducted using RGB images, while this method was originally designed for grayscale samples only.

[1] Uncompressed version of images shown in this paper are available at https://tinyurl.com/icmlade

Dataset		CelebA				UC Merced Landuse			
Noise level	Model	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
0.04	Ours (ODE)	**13.9156**	**0.0365**	31.8902	0.9348	**35.7451**	**0.0753**	31.2300	0.9137
	DeblurGAN	18.7211	0.0545	29.6118	0.9274	50.6455	0.1291	29.7493	0.8971
	Restormer	21.9359	0.0523	**34.1125**	**0.9664**	51.3572	0.1043	**32.7498**	**0.9377**
	MPRNet	23.0067	0.0544	34.0028	0.9654	64.2903	0.1176	32.5545	0.9352
	NAFNet	20.5896	0.0503	34.0764	0.9661	51.4628	0.1063	32.4929	0.9322
	DnCNN	26.8650	0.0726	33.1393	0.9542	111.3414	0.1628	31.8040	0.9217
	SRAD	47.7516	0.2374	27.5801	0.8476	70.5202	0.3565	27.7434	0.8318
	CBM3D	25.1978	0.1282	29.5931	0.9067	74.0214	0.2553	29.8244	0.8836
0.12	Ours (ODE)	**24.3077**	**0.0774**	28.5567	0.8994	**63.2679**	**0.1333**	28.8183	0.8727
	DeblurGAN	38.3176	0.0902	24.3920	0.8360	93.9357	0.1916	26.1810	0.8218
	Restormer	29.6475	0.0848	**31.5912**	**0.9473**	76.1931	0.1653	**30.2727**	**0.9016**
	MPRNet	31.5731	0.0906	31.4240	0.9447	98.2022	0.1884	30.0573	0.8965
	NAFNet	27.0304	0.0805	31.5588	0.9466	77.2864	0.1644	30.0889	0.8977
	DnCNN	33.2722	0.1113	30.5007	0.9281	138.2030	0.2361	29.2134	0.8710
	SRAD	70.4303	0.3853	24.3354	0.7311	107.6315	0.4712	25.2329	0.7247
	CBM3D	36.2575	0.2128	26.1749	0.8481	100.1406	0.3570	27.1042	0.8144

TABLE II
COMPARISON OF DIFFERENT DENOISING METHODS FOR VARIOUS METRICS AT DIFFERENT NOISE LEVELS ON CELEBA AND UC MERCED LANDUSE DATASET. THE BEST PERFORMANCE FOR EACH METRIC IS HIGHLIGHTED IN BOLD.

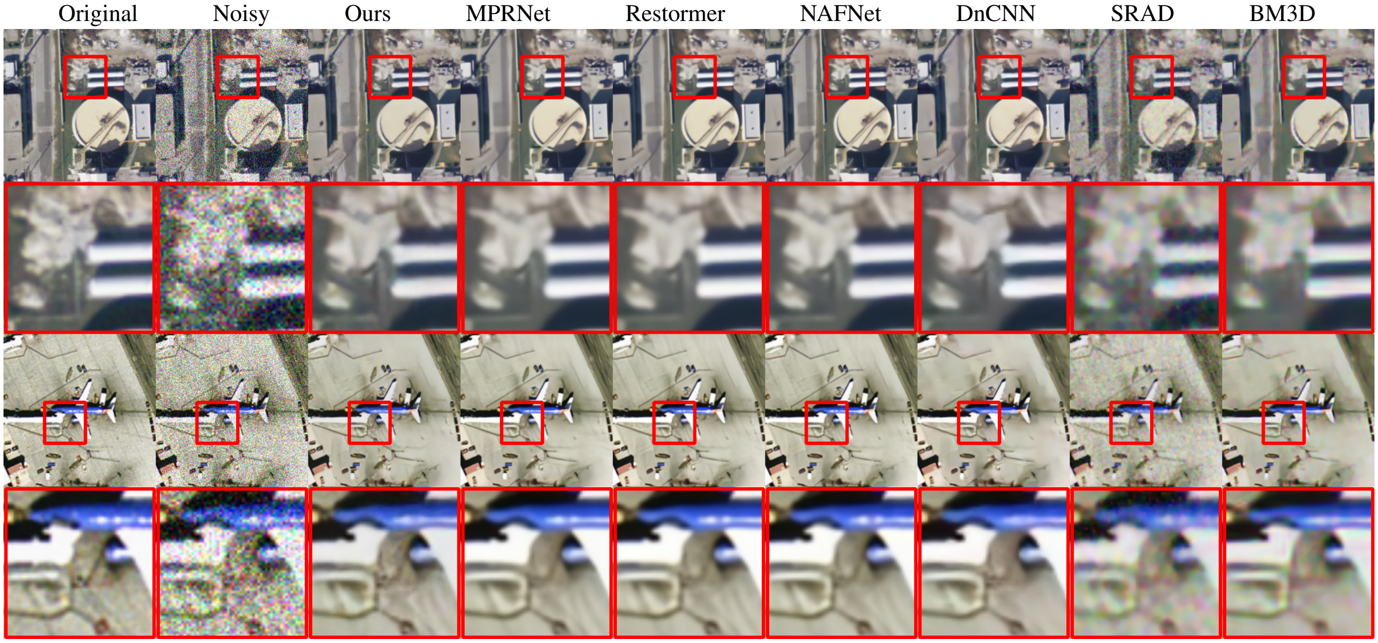

Fig. 4. Comparing between different denoising models on randomly selected LandUse images, at noise level 0.12.

For DeblurGAN, while it performs fine at low noise level, it gets substantially worse than other methods across all metrics when more noise is present in the images. In either case, our method also beats it decisively.

These observations can also be seen in Figure 3, where we present denoising results of theses models at noise level 0.12. Compared to our method, other techniques suffer from over-smoothing and they usually generate samples that lack high-frequency details. This is the drawback of using PSNR as an optimization objective, where the denoisers have the tendency of collapsing into the mean statistics of the images, creating smoothing effect. In contrast, our proposed model tends to be much better at preserving finer details, such as facial hair and clothes wrinkles.

Out-of-distribution results on Land Use dataset. To evaluate the generalization capability of our method, we re-run the previous experiments on the UC Merced Land Use dataset, which is a small dataset (containing 2,096 samples) captured from satellites. Besides being out-of-distribution, we note that this is a much harder dataset to do denoising on, due to the blurriness and color shifting of satellite imaging. We present quantitative results in the right half of Table II. While there is large degradation in performance across the board, we again see that our method achieves the best FID and LPIPS scores, while being slightly worse in terms of PSNR and SSIM. Surprisingly, DnCNN performs much worse in terms of FID, while remaining competitive in the other metrics.

Qualitatively, from Figure 4, it is observed that our method retains finer details of the images. DnCNN, NAFNet, MPRNet, Restormer, and DeblurGAN produce over-smoothed samples, losing details of the ground, while SRAD and BM3D generate images that completely lack high-frequency components.

Overall, we see that our method achieves competitive performance on PSNR and SSIM, while producing more realistic samples that are closer to the true input distribution, as measured by FID and LPIPS metrics. Furthermore, we show empirically that the method can generalize well to out-of-distribution dataset, which is crucial in applications where data samples are limited.

V. Conclusion

In conclusion, this paper introduces a novel SDE-based diffusion model for removing multiplicative noise. The work presents the construction of the forward and reverse SDEs that directly captures the dynamics of the noise model. In addition, it also establishes the training objective as well as multiple different sampling equations based on Probability flows and DDIM techniques. The proposed model is compared to classical image processing algorithms, including BM3D and SRAD, as well as the modern CNN-based methods. Extensive experiments on different datasets demonstrate 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.

Going forward, we will explore the application of diffusion steps reduction techniques to noise removal problems. While these techniques have been applied successfully in generative tasks, it greatly reduces the quality of generated samples in our problem. Thus, care need to be taken when dealing with tasks that are sensitive to small perturbation like denoising. Furthermore, while deterministic sampling is usually used to speedup generation, it is a desirable property in noise removal tasks. Specifically, we would like the process to produce the exact clean image, not something close in terms of distribution, which is modeled by the current diffusion loss. This has connections to conditioning using Doob's h-transform, and could hold interesting research venue, we leave it to future works.

References

[1] Yongjian Yu and Scott T Acton, "Speckle reducing anisotropic diffusion," *IEEE Transactions on image processing*, vol. 11, no. 11, pp. 1260–1270, 2002.

[2] Jianing Shi and Stanley Osher, "A nonlinear inverse scale space method for a convex multiplicative noise model," *SIAM Journal on imaging sciences*, vol. 1, no. 3, pp. 294–321, 2008.

[3] Gilles Aubert and Jean-Francois Aujol, "A variational approach to removing multiplicative noise," *SIAM journal on applied mathematics*, vol. 68, no. 4, pp. 925–946, 2008.

[4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[5] 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*, vol. 26, no. 7, pp. 3142–3155, 2017.

[6] Dazi Li, Wenjie Yu, Kunfeng Wang, Daozhong Jiang, and Qibing Jin, "Speckle noise removal based on structural convolutional neural networks with feature fusion for medical image," *Signal Processing: Image Communication*, vol. 99, pp. 116500, 2021.

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

[8] Danlei Feng, Weichen Wu, Hongfeng Li, and Quanzheng 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*. Springer, 2020, pp. 85–92.

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

[10] Tongda Yang, Weiming Wang, Gary Cheng, Mingqiang Wei, Haoran Xie, and Fu Lee Wang, "Fddl-net: frequency domain decomposition learning for speckle reduction in ultrasound images," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 42769–42781, 2022.

[11] Jing Zhang, Wenguang Li, and Yunsong Li, "Sar image despeckling using multiconnection network incorporating wavelet features," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 8, pp. 1363–1367, 2019.

[12] Gang Liu, Hongzhaoning Kang, Quan Wang, Yumin Tian, and Bo Wan, "Contourlet-cnn for sar image despeckling," *Remote Sensing*, vol. 13, no. 4, pp. 764, 2021.

[13] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun, "Simple baselines for image restoration," in *Computer Vision – ECCV 2022*, 2022, pp. 17–33.

[14] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao, "Multi-stage progressive image restoration," in *CVPR*, 2021.

[15] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *CVPR*, 2022.

[16] Soumee Guha and Scott T Acton, "Sddpm: Speckle denoising diffusion probabilistic models," *arXiv preprint arXiv:2311.10868*, 2023.

[17] Malsha V Perera, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M Patel, "Sar despeckling using a denoising diffusion probabilistic model," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[18] Siyao Xiao, Libing Huang, and Shunsheng Zhang, "Unsupervised sar despeckling based on diffusion model," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023, pp. 810–813.

[19] Naama Pearl, Yaron Brodsky, Dana Berman, Assaf Zomet, Alex Rav Acha, Daniel Cohen-Or, and Dani Lischinski, "Svnr: Spatially-variant noise removal with denoising diffusion," *arXiv preprint arXiv:2306.16052*, 2023.

[20] Simo Särkkä and Arno Solin, *Applied stochastic differential equations*, vol. 10, Cambridge University Press, 2019.

[21] Aapo Hyvärinen and Peter Dayan, "Estimation of non-normalized statistical models by score matching.," *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.

[22] Jiaming Song, Chenlin Meng, and Stefano Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[24] Pascal Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[25] HH Arsenault and G April, "Properties of speckle integrated with a finite aperture and logarithmically transformed," *JOSA*, vol. 66, no. 11, pp. 1160–1163, 1976.

[26] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[27] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas, " DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks ," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.