2025 IEEE International Conference on Image Processing (ICIP) | 979-8-3315-2379-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICIP55913.2025.11084619

DEEP UNSUPERVISED DESPECKLING WITH UNBIASED RISK ESTIMATION

Ashutosh Gupta[1,4] *Chandra Sekhar Seelamantula*[2] *Thierry Blu*[3] *Nitant Dube*[1] *Shanmuganathan Raman*[4]

[1] Space Applications Centre, ISRO, Ahmedabad, India
[2] Department of Electrical Engineering, Indian Institute of Science (IISc.), Bengaluru, India
[3]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[4] Indian Institute of Technology (IIT), Gandhinagar, India

ABSTRACT

Despeckling of Synthetic Aperture Radar (SAR) images has seen significant progress in recent years, largely driven by advancements in deep learning techniques. However, many of these approaches face challenges when applied to new SAR datasets, primarily due to their dependence on ground truth images, which are often unavailable for real-world sensors. In this paper, we address this limitation by extending the concept of unbiased risk estimation in the presence of Gamma-distributed multiplicative speckle. Specifically, we demonstrate that it is possible to train deep denoising networks without relying on ground truth data using our estimator. We introduce a new formulation of the Multiplicative Unbiased Risk Estimator (MURE) and present a computationally efficient Monte Carlo-based method that enables accurate estimation of the modified MURE cost, facilitating effective unsupervised training of deep neural networks from large datasets consisting solely of noisy SAR images. Experimental results on both synthetic datasets and real Sentinel-1 SAR images validate the suitability of our method for real-world applications. Even without ground truth, our method achieves performance that closely matches the Oracle-based denoiser and proves superior to the out-of-domain performance of popular supervised SAR despeckling methods.

***Index Terms*—** Despeckling, SAR, multiplicative noise, deep learning, MURE.

1. INTRODUCTION

Active imaging sensors such as SAR play a pivotal role in various applications, including remote sensing, medical imaging, and security systems. Unlike passive sensors, which rely on ambient light or natural radiation, active imaging systems emit signals—such as laser or coherent pulses—and capture the reflected signal to construct images. This self-illuminating capability makes them particularly effective for Earth observation studies due to their ability to penetrate clouds and operate at night. However, these systems are often severely affected by *speckle* - a multiplicative noise that arises from the interference of coherent waves scattered by the imaged objects [1, 2]. These fluctuations seriously affect human and automated interpretation of these images. A denoising process called *despeckling* is often applied to alleviate this. The statistical model [1] for fully developed speckle provides the key to understanding speckle and its properties. For a SAR system, it is possible to factor the reflected complex amplitude into two parts- one with only the deterministic but unknown reflectance to be measured ($\mathbf{x} \in \mathbb{R}_+^m$) and the other as the unwanted signal-dependent speckle ($\mathbf{N} \in \mathbb{R}_+^m$) caused by constructive and destructive interference of the random scatters. The noisy measurements $\mathbf{Y} \in \mathbb{R}_+^m$ follow the model:

$$\mathbf{Y} = \mathbf{x} \odot \mathbf{N}, \tag{1}$$

where $\odot$ is the Hadamard product, images are treated as m-dimensional vector random variables. For fully developed *multilook* speckle with k looks, the components of random variable $\mathbf{N}$ are i.i.d. and follow a Gamma distribution with mean $\mu_N = 1$ and variance $\sigma_N^2 = 1/k$:

$$p_N(n; k, k) = \frac{k^k}{\Gamma(k)}\, n^{k-1} e^{-kn}\; n > 0, k > 0, \tag{2}$$

where $\Gamma(.)$ is the Gamma function. To demonstrate the severity of speckle, a single-look Sentinel-1 SAR image and despeckling result from our unsupervised method are shown in Figure 1 for reference.

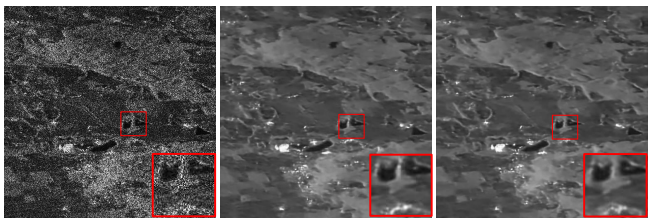

Fig. 1: Sentinel-1 SAR image, unsupervised DeepMURE (center) performs close to the Oracle (right).

1.1. Prior work

Early despeckling methods used additive denoising techniques by applying homomorphic filtering to transform the

noise model into an additive form. Other approaches, like the Kuan [3] and Lee filters [4], relied on local statistics but struggled to preserve fine details due to oversimplified assumptions. These limitations were partly addressed by total variation denoising [5] and maximum a *posteriori* (MAP) methods with regularization [6]. More recently, non-local and collaborative filtering techniques, such as NL-SAR [7] and SAR-BM3D [8], have been applied to SAR despeckling yielding superior results.

Supervised deep-learning methods. With the advances in data-driven techniques for image restoration, numerous supervised deep learning methods have been developed for despeckling. Notably, SAR-CNN methods [9, 10, 11] train a Convolutional Neural Network (CNN) in a homomorphic setting and utilize residual learning to obtain denoised images. Other methods such as SAR-DRN[12] and MO-Net [13] design specialized architectures and utilize total-variation, KL-divergence, and despeckling gain (DG) losses apart from the commonly used Mean-Squared Error (MSE) to improve the visual quality of recovered outputs. Recently, authors of SAR-Transformer [14] utilized the global attention capability of transformer architecture for the despeckling task. A major issue with these methods is the sufficient availability of ground truth images, which are rarely available. Consequently, methods trained on synthetically generated datasets fail to generalize to unseen image domains.

Self-supervised learning methods. To address the limitations of supervised methods, self-supervised approaches leverage supervisory signals derived directly from the noisy image. In this context, SAR2SAR [15] adopts the *Noise2Noise* framework to perform denoising by utilizing multi-temporal images of the same region. Similarly, MERLIN [16] builds on this concept but applies the technique to the real and complex components of Single Look Complex (SLC) images. However, these methods require complex images and specific information about the system, which may not always be accessible.

1.2. Our Contributions

We draw inspiration from prior work on unsupervised denoising for additive Gaussian noise. At the core of our approach is the use of unbiased risk estimators, which allow estimation of Mean-Squared-Error (MSE) in the absence of ground truth. Works such as [17, 18] demonstrate that Stein's Unbiased Risk Estimator (SURE) is an effective surrogate of MSE for Gaussian denoising. Such estimators also exist for Poisson noise [19]. Building on the work of [20], which introduces a Multiplicative Unbiased Risk Estimator (MURE), we:

1. Introduce a novel expression for MURE, consisting solely of derivative terms, in contrast to the original MURE, which includes a difficult-to-evaluate integral term (Corollary 1.1).
2. Develop a computationally efficient Monte Carlo method using only input image perturbations and demonstrate its application in training a deep neural network (DnCNN [21]).
3. Conduct experiments on both real and synthetic images in an unsupervised training regime with DeepMURE (Deep despeckling with MURE), enabling training on datasets containing only noisy images.

Our unsupervised method achieves PSNR performance within 1 dB of the denoiser trained with true MSE (Oracle).

2. UNSUPERVISED DESPECKLING WITH MURE

2.1. Theory

2.1.1. Unbiased Risk Estimation

Most image restoration methods utilize the MSE loss (*risk*) apart from some specialized loss functions. The MSE depends on the ground truth samples $\mathbf{x}$ that are not easy to obtain. To enable estimation of MSE without ground truth, Seelamantula and Blu [20] introduced MURE for Gamma distributed multiplicative noise and derived a linear, wavelet-based denoiser. We restate MURE in its multivariate (vector) form and derive a new version that is computationally superior to the original MURE version.

Theorem 1. *(Multivariate version) Let $\mathbf{Y} = \mathbf{xN}$, where $\mathbf{x} \in \mathbb{R}_+^{\mathbf{m}}$ is deterministic but unknown reflectance image. Let $\mathbf{Y}, \mathbf{N} \in \mathbb{R}_+^m$, and $\mathbf{N} \sim \Gamma(k,k)$ with independent entries, then, the vector random variable*

$$\hat{\zeta}(\mathbf{f}) = \frac{k}{k+1}\|\mathbf{Y}\|^2 - 2\mathbf{Y}^T\mathcal{M}\mathbf{f}(\mathbf{Y}) + \|\mathbf{f}(\mathbf{Y})\|^2 \qquad (3)$$

is an unbiased estimator of the MSE, $\zeta(\mathbf{f}) = \mathbb{E}_{\mathbf{N}}\{\|\mathbf{f}(\mathbf{Y}) - \mathbf{x}\|^{\mathbf{2}}\}$, where $\mathbb{E}$ is the expectation operator. For a scalar function $f(Y)$, the operator $\mathcal{M}$ is defined as $\mathcal{M}f(Y) = k\int_0^1 s^{k-1}f(sY)ds$. This notation is extended straightforwardly to multivariate vector functions $\mathbf{f}(\mathbf{Y}) = [f_1(\mathbf{Y}), f_2(\mathbf{Y}), \ldots, f_m(\mathbf{Y})]^{\mathrm{T}}$ according to $\mathcal{M}\mathbf{f}(\mathbf{Y}) = [\mathcal{M}_1 f_1(\mathbf{Y}), \mathcal{M}_2 f_2(\mathbf{Y}), \ldots, \mathcal{M}_m f_m(\mathbf{Y})]^{\mathrm{T}}$, where $\mathcal{M}_i\mathbf{f_i}(\mathbf{Y})$ applies the operator $\mathcal{M}$ to the i^{th} input component of $\mathbf{f}(\mathbf{Y})$ only.

For the proof, see [20]. While the MURE estimator ($\hat{\zeta}(\mathbf{f})$) can be used as a surrogate to MSE, the presence of integral in the definition of $\mathcal{M}$ makes it computationally demanding. Specifically, the original MURE estimate has an exponential cost in the number of pixels m and the number of parameters of the denoiser, $|\theta|$. Moreover, since it is difficult to solve exactly for non-linear integrands, numerical integration methods need to be applied, which are time-consuming even for the simplest of the denoisers (e.g., soft thresholding in the wavelet domain). To give an example, the original MURE

cost would require ∼100 K denoiser evaluations even for a modestly sized 512×512 SAR image. Due to this, MURE is computationally prohibitive to evaluate, especially for deep neural networks with millions of parameters. To alleviate this, we derive a new version of MURE and devise a Monte Carlo (MC) estimation technique that can be used to train very deep neural networks efficiently without supervision.

Corollary 1.1. ***(Series version)*** *An equivalent expression for cost $\hat{\zeta}(\mathbf{f})$ in Eq. (3) for a differentiable denoiser $\mathbf{f}(\mathbf{Y})$ is*

$$\hat{\zeta}(\mathbf{f}) = \frac{k}{k+1}\|\mathbf{Y}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 - 2\sum_{i=1}^{m}\sum_{p=0}^{\infty}(-1)^p \frac{k!}{(k+p)!} Y_i^{p+1} \frac{\partial \mathbf{f}_i^{(p)}(\mathbf{Y})}{\partial Y_i}, \quad (4)$$

where p is the order of partial derivative of $\mathbf{f}_i(\mathbf{Y})$ w.r.t. $\mathbf{Y}_i$, the i^{th} pixel of the input noisy image.

Proof. Since only the term $\mathbf{Y}^T\mathcal{M}\mathbf{f}(\mathbf{Y})$ in Theorem 1 is different from this version, we focus on its expansion. We rewrite the component-wise operator $\mathcal{M}_i f_i(\mathbf{Y})$ by applying integration by parts:

$$\begin{aligned}
\mathcal{M}_i f_i(\mathbf{Y}) &= k\int_0^1 s^{k-1} f_i(Y_1, Y_2, ..., sY_i, ..., Y_m)\mathrm{d}s \\
&= k\int_0^1 s^{k-1} f_i(\mathbf{S}_i\mathbf{Y})\mathrm{d}s \\
&= s^k f_i(\mathbf{S}_i\mathbf{Y})\big|_0^1 - \int_0^1 Y_i f_i^{'}(\mathbf{S}_i\mathbf{Y}) s^{k+1}\mathrm{d}s, \\
&= f_i(\mathbf{Y}) - \int_0^1 Y_i f_i^{'}(\mathbf{S}_i\mathbf{Y}) s^{k+1}\mathrm{d}s.
\end{aligned}$$

where, we introduce $\mathbf{S}_i \in \mathbb{R}^{m\times m}$ as a matrix constructed by replacing i^{th} diagonal element of identity matrix by s. Applying integration by parts repeatedly to this expression yields:

$$\mathcal{M}_i f_i(\mathbf{Y}) = \sum_{p=0}^{\infty}(-1)^p \frac{k!}{(k+p)!} Y_i^p \frac{\partial f_i^{(p)}(\mathbf{Y})}{\partial Y_i}.$$

For the vector version of the estimate, we need to evaluate the inner product of this term with $\mathbf{Y}$, which gives:

$$\mathbf{Y}^T\mathcal{M}\mathbf{f}(\mathbf{Y}) = \sum_{i=1}^{m}\sum_{p=0}^{\infty}(-1)^p \frac{k!}{(k+p)!} Y_i^{p+1} \frac{\partial f_i^{(p)}(\mathbf{Y})}{\partial Y_i},$$

which completes the proof. □

2.1.2. Monte Carlo estimation of MURE

Compared to the expression in the original MURE requires m^2 evaluations of the integral, the cross-product term $\mathbf{Y}^T\mathcal{M}\mathbf{f}(\mathbf{Y})$ in Corollary 1.1 contains only inner product of the noisy image and the diagonal elements of the partial derivatives. While it is possible to utilize *automatic differentiation* capabilities of modern deep learning frameworks to compute the terms of our version cost, obtaining full ($\mathcal{O}(m^{p+1})$)-sized derivative tensors is computationally inefficient as we only need the *diagonal* entries of p^{th} derivative for taking the inner product with $\mathbf{Y}^{p+1}$. Instead, we propose a Monte Carlo approach and show that it is possible to estimate the inner product directly via ensemble averages of finite difference operations applied to the denoiser's response to specifically crafted input perturbations. To see this, first, we explicitly write the terms $p \in \{0, 1, 2\}$ in Eq. 4:

$$\hat{\zeta}(\mathbf{f}) = \frac{k}{k+1}\|\mathbf{Y}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 - 2\sum_{i=1}^{m}\left(Y_i\mathbf{f_i}(\mathbf{Y}) - \frac{Y_i^2[\mathbf{J}_f(\mathbf{Y})]_{\mathbf{ii}}}{k+1} + \frac{Y_i^3[\mathbf{H}_f(\mathbf{Y})]_{\mathbf{iii}}}{(k+1)(k+2)} + ...\right) \quad (5)$$

where $[\mathbf{J}_f(\mathbf{Y})]_{\mathbf{ii}}$ is i^{th} diagonal entry of the Jacobian of $\mathbf{f}(\mathbf{Y})$ and $[\mathbf{H}_f(\mathbf{Y})]_{\mathbf{iii}}$ is the i^{th} diagonal entry of the Hessian of $\mathbf{f_i}(\mathbf{Y})$. The product with Y_i^{p+1} terms can be seen as a weighted trace of the p^{th} derivative. The term for $p = 0$ is easy to evaluate as an inner product between the noisy and denoised images. We now present our two key results - Theorem 2 and Theorem 3, which enable the efficient Monte Carlo estimation of MURE terms for $p = 1$ and $p = 2$, respectively.

Theorem 2. *Let $\mathbf{f}(\mathbf{Y}) : \mathbb{R}^{\mathbf{m}} \to \mathbb{R}^{\mathbf{m}}$ be a differentiable function and $\mathbf{J_f}$ denote its Jacobian. Let $\mathbf{B} \in \mathbb{R}^{\mathbf{m}}$ with i.i.d entries $\sim \mathcal{N}(0, 1)$. Then, $\sum_{i=1}^{m}[\mathbf{J_f}(\mathbf{Y})]_{ii}Y_i^2 =$ $\lim_{\epsilon\to 0}\mathbb{E}_B\left\{(\mathbf{Y}\odot\mathbf{B})^{\mathbf{T}}\left(\frac{\mathbf{f}(\mathbf{Y}+\epsilon(\mathbf{Y}\odot\mathbf{B}))-\mathbf{f}(\mathbf{Y}-\epsilon(\mathbf{Y}\odot\mathbf{B}))}{\mathbf{2}\epsilon}\right)\right\}$.*

A similar approximation can be obtained for $p = 2$ term.

Theorem 3. *Let $\mathbf{f}(\mathbf{Y}) : \mathbb{R}^{\mathbf{m}} \to \mathbb{R}^{\mathbf{m}}$ be at a twice differentiable function and $[\mathbf{H_f}]_{\mathbf{i}}$ denote the Hessian of $\mathbf{f}_i$. Let $\mathbf{B} \in \mathbb{R}^{\mathbf{m}}$ with i.i.d entries $\sim Triangular(-2, 1)$. Then $\mathbb{E}_B\left\{-5(\mathbf{Y}\odot\mathbf{B})^{\mathbf{T}}\left(\frac{\mathbf{f}(\mathbf{Y}+\epsilon(\mathbf{Y}\odot\mathbf{B}))-\mathbf{2f}(\mathbf{Y})+\mathbf{f}(\mathbf{Y}-\epsilon(\mathbf{Y}\odot\mathbf{B}))}{\epsilon^{\mathbf{2}}}\right)\right\}$ estimates $\sum_{i=1}^{m}[\mathbf{H_f}(\mathbf{Y})]_{iii}Y_i^3$ in the limit as $\epsilon \to 0$.*

The proofs for Theorem 2 and Theorem 3 are based on approximation of the derivatives using a Taylor series expansion. Due to space limitations, we plan to provide the detailed proofs of these theorems in a future journal version of the work. The expectation $\mathbb{E}_{\mathbf{B}}$ for the terms are computed using Monte Carlo estimates with K perturbations of the input noisy images. Since our version of MURE uses only finite difference operations, evaluation of the term $\mathbf{Y}^T\mathcal{M}\mathbf{f}(\mathbf{Y})$ becomes not only tractable but rather cheap compared to numerical integration or even *autograd* based estimation.

We estimate only the first three terms ($p \leq 2$) as higher-order terms increase computational complexity with minimal benefit. Figure 2 shows that with $p = 0$, the network fits the noisy image directly, increasing weighted divergence, defined as the sum of terms for $p = 1$ and $p = 2$. For $p \leq 1$, the MSE

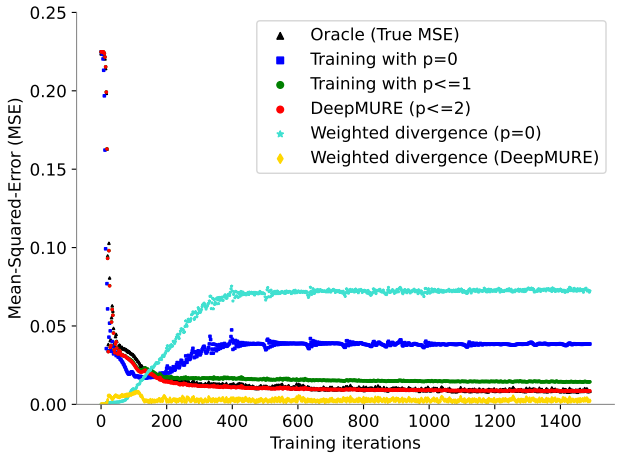

Fig. 2: Effect of number of terms (p) of the MURE cost on MSE estimation during neural network training.

approaches the oracle MSE, while for $p \leq 2$ (DeepMURE), the cost remains close to the oracle cost throughout training, as confirmed by the low-weighted divergence. We empirically set $\epsilon = 1\text{E}{-2}$ and observe that a single perturbation ($K = 1$) suffices due to the high dimensionality of images. Using the series version of MURE, only four additional network evaluations per iteration are needed, compared to $\mathcal{O}(m^2)$ operations in the original cost. Thus, all experiments use the final cost in Eq. (5) with $p \leq 2$ and $K = 1$.

2.2. Implementation Details

We implement MURE as a plug-and-play loss in Tensorflow. To demonstrate our method, we choose DnCNN [21] as a representative denoiser for its proven denoising capability. To train a deep neural network $\mathbf{f}_\theta$ with parameters as a random variable Θ, we minimize the following objective:

$$\Theta^* = \arg\min_{\Theta} \text{MURE}(\mathbf{f}_\Theta(\mathbf{Y})) \tag{6}$$

over the noisy realizations of $\mathbf{Y}$ in the dataset. We compare the performance of DeepMURE with the Oracle-MSE performance and some other state-of-the-art despeckling methods. To estimate the MURE cost, we require several looks (k) of the noisy image, which is either assumed known or estimated using the statistical estimation technique using homogeneous regions as described in [2]. We train the network with a learning rate of 1E-5 with an ADAM optimizer for 50 epochs. Computationally, every MSE estimation step involves $4K$ network evaluations for the Jacobian and Hessian terms compared to 1 evaluation for the Oracle-MSE, where K is the number of perturbations. Hence, the training time for DeepMURE is 4 times that of oracle-MSE. A DeepMURE trained DnCNN takes 0.4 seconds for despeckling a 512×512 image in comparison to 600 seconds required by SAR-BM3D. All our experiments are performed on a single Nvidia-V100 GPU using the TensorFlow deep learning framework.

Table 1: Quantitative comparison of methods on Sentinel-1 SAR and UC-Merced dataset with single-look speckle. Best results are in bold-face and second best results are in italics.

	Sentinel-1		UC-Merced	
Method	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Noisy ($k = 1$)	14.46	0.27	11.90	0.28
SAR-BM3D [8]	26.11	*0.80*	22.37	0.57
SAR-CNN [9]	25.46	0.70	23.46	*0.62*
SAR-Transformer [14]	27.93	0.77	23.85	0.59
DeepMURE	*28.16*	0.79	*24.12*	*0.62*
Oracle	**28.88**	**0.82**	**24.78**	**0.65**

3. EXPERIMENTS

We conduct experiments on the UC-Merced [22] and Sentinel-1 SAR [10] datasets to demonstrate our method's effectiveness. The UC-Merced dataset contains high-resolution images of 21 land classes, with 100 images per class. We create a test set of 21 representative images, adding synthetic speckles with $k = 1$, corresponding to the highest noise level. For Sentinel-1, we train DeepMURE on single-look images and compare results with the multi-temporal processed images provided in the dataset. Performance is measured using the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). DeepMURE trains DnCNN using only noisy images, while the Oracle uses ground truth references. Although our framework can train any linear or non-linear denoiser, we compare it with some baseline methods with different architectural backbones. We select SAR-BM3D, a widely used training-free method, and two recent supervised deep learning methods SAR-CNN [9] and SAR-Transformer [14] for the comparison. To highlight the domain adaptation challenge in supervised methods, we use the original versions of these models without retraining on the target datasets. For facilitating comparison with a supervised method trained on the same dataset, Oracle performance is compared both visually and quantitatively.

Figure 3 provides a visual and quantitative comparison of one sample each from synthetic and real datasets, with a full evaluation presented in Table 1. For both samples, SAR-BM3D improves over the noisy input but introduces significant artifacts in high speckle regions. SAR-CNN produces sharper outputs with better PSNR and SSIM values than SAR-BM3D, but it introduces noticeable artifacts, especially near boundaries. Similarly, the SAR-Transformer struggles with smudged outputs due to the lack of dataset-specific training. In contrast, DeepMURE, trained directly on noisy realizations without ground truth, significantly outperforms SAR-CNN on the Sentinel dataset, achieving a 2.5 dB gain in PSNR and 0.09 in SSIM. On the synthetically corrupted UC-Merced dataset, DeepMURE achieves smaller but consistent gains of 0.7 dB and 0.3 dB in PSNR over SAR-CNN and SAR-Transformer, respectively. These more significant im-

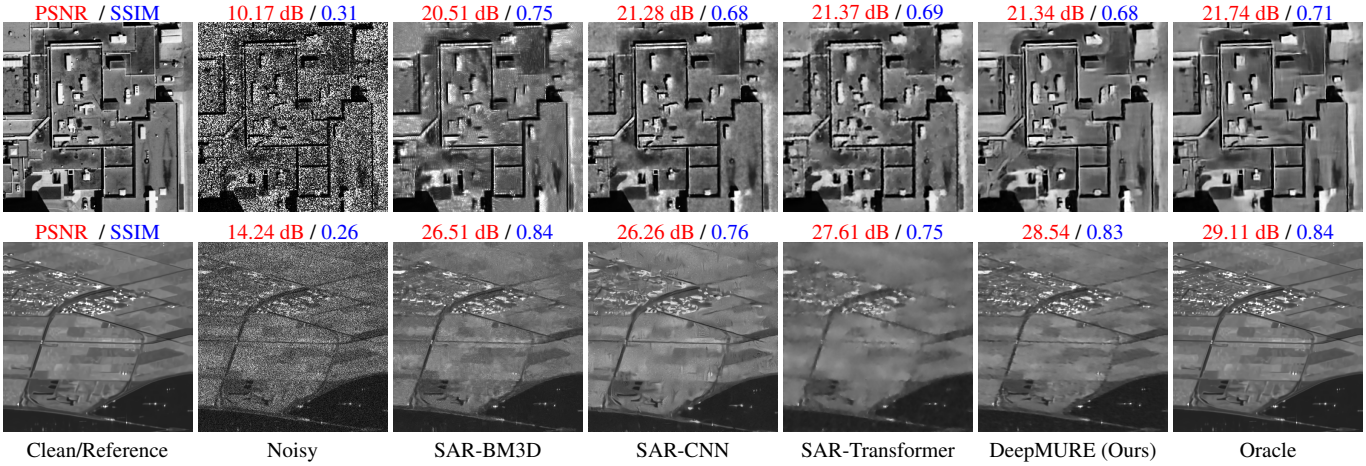

Fig. 3: Visual and quantitative comparison for UC-Merced (synthetic) and Sentinel-1 (SAR) samples.

provements on real datasets highlight DeepMURE's ability to adapt to target distributions effectively. By leveraging same-domain training, DeepMURE produces visually superior outputs compared to SAR-CNN and SAR-Transformer, demonstrating its robustness and adaptability.

DeepMURE performs slightly below the Oracle-trained network, with an average PSNR drop of 0.7 dB and a 0.03 reduction in SSIM. While visually close to the Oracle outputs, DeepMURE exhibits higher residual noise in smooth regions, primarily due to the absence of ground-truth supervision with the MSE-only loss. This limitation could be addressed by incorporating unsupervised losses, such as TV loss or perceptual loss, into the training objective. Despite this, DeepMURE achieves results that are remarkably close to the Oracle, even with a single perturbation setting for Monte Carlo estimation.

3.1. Ablation Studies

To evaluate the estimation performance of the series version of MURE, we vary the number of terms ($p \in 0, 1, 2$), the number of perturbations for Monte Carlo estimation ($K \in 1, 12, 24$), and the number of looks ($k \in 1, 4, 10$). We train a DnCNN on 7 sample images of size 512×512 from the Sentinel-1 test dataset and perform three network training per setting, reporting the mean and standard deviation of PSNR estimates compared to Oracle PSNR. We also report average run times for different perturbation counts. As shown in Table 2, increasing K from 1 to 24 reduces the MURE estimation standard deviation by up to 0.3 dB with $p = 2$ and $k = 1$. Settings with $p < 2$ underperform, highlighting the importance of including higher-order terms in the cost estimate. As expected, higher numbers of looks improve performance, with MURE coming within 0.2 dB of the Oracle at $k = 10$. While increasing perturbations enhance PSNR estimates, it also significantly increases computational cost, justifying our choice of $K = 1$ for all experiments.

Table 2: PSNR estimation gap (dB) compared to oracle and run-time (100 iterations) comparison for different settings.

(K)	Number of MURE terms (p)			Number of looks (k)			Time (s)
	0	**1**	**2**	**1**	**4**	**10**	
1	3.72 ± 2.6	1.37 ± 0.64	1.21 ± 0.57	1.23 ± 0.57	0.74 ± 0.31	0.21 ± 0.12	33
12	3.91 ± 1.3	1.41 ± 0.37	1.17 ± 0.33	1.17 ± 0.33	0.71 ± 0.17	0.23 ± 0.03	391
24	3.62 ± 0.9	1.45 ± 0.21	1.13 ± 0.21	1.13 ± 0.21	0.65 ± 0.12	0.22 ± 0.03	727

4. CONCLUSIONS

This work presents an alternate formulation of the Multiplicative Unbiased Risk Estimator (MURE) that relies exclusively on derivative terms. This new form allows the MURE cost to be efficiently approximated for any denoiser using a Monte Carlo approach, eliminating the need for computationally expensive numerical methods involving integrals. Moreover, this formulation enables the training of non-linear denoisers, such as deep neural networks, without ground truth. We make unsupervised training practical and effective by leveraging input perturbations and simple difference operations, which are computationally efficient on modern hardware. Experimental results demonstrate that a deep denoising network trained unsupervised with MURE achieves performance comparable to Oracle, even on single-look SAR observations. These findings validate the effectiveness of our proposed unsupervised despeckling framework and highlight its potential for real-world applications. In the future, we would like to extend our work to a more general SAR imaging model.

5. ACKNOWLEDGMENTS

Authors would like to acknowledge the encouragement received from Director, SAC as well as all the colleagues at SIPA, SAC (ISRO). Authors also acknowledge the Jibaben Patel Chair in Artificial Intelligence, IIT, Gandhinagar and ISRO-IISc. Space Technology Cell for the support received.

T.B. acknowledges funding support from the Yushan Fellow Program (MOE, Taiwan).

6. REFERENCES

[1] J. W. Goodman, "Some fundamental properties of speckle," *JOSA*, vol. 66, no. 11, pp. 1145–1150, 1976.

[2] F. T. Ulaby, F. Kouyate, B. Brisco, and T. L. Williams, "Textural infornation in sar images," *IEEE Transactions on Geoscience and Remote Sensing*, , no. 2, pp. 235–245, 1986.

[3] D. Kuan, A. Sawchuk, T. Strand, and P. Chavel, "Adaptive restoration of images with speckle," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 373–383, 1987.

[4] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE transactions on pattern analysis and machine intelligence*, , no. 2, pp. 165–168, 1980.

[5] S. Osher, N. Paragios, L. Rudin, P.-L. Lions, and S. Osher, "Multiplicative denoising and deblurring: theory and algorithms," 2003.

[6] J. M. Bioucas-Dias and M. A. Figueiredo, "Multiplicative noise removal using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1720–1730, 2010.

[7] C.-A. Deledalle, L. Denis, F. Tupin, A. Reigber, and M. Jäger, "Nl-sar: A unified nonlocal framework for resolution-preserving (pol)(in) sar denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2021–2038, 2014.

[8] S. Parrilli, M. Poderico, C. V. Angelino, and L. Verdoliva, "A nonlocal sar image denoising algorithm based on llmmse wavelet shrinkage," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 606–616, 2011.

[9] P. Wang, H. Zhang, and V. M. Patel, "Sar image despeckling using a convolutional neural network," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1763–1767, 2017.

[10] E. Dalsasso, X. Yang, L. Denis, F. Tupin, and W. Yang, "Sar image despeckling by deep neural networks: from a pre-trained model to an end-to-end training strategy," *ArXiv*, vol. abs/2006.15559, 2020.

[11] C.-A. Deledalle, L. Denis, and F. Tupin, "Speckle reduction in matrix-log domain for synthetic aperture radar imaging," *Journal of Mathematical Imaging and Vision*, vol. 64, no. 3, pp. 298–320, 2022.

[12] Y. Zhang, X. Xing, D. Meng, Y. Li, and L. Zhang, "Learning a Dilated Residual Network for SAR Image Despeckling," *Remote Sensing*, vol. 10, no. 2, pp. 196, 2018.

[13] S. Vitale, H. Aghababaei, G. Ferraioli, V. Pascazio, and G. Schirinzi, "A multi-objective approach for multichannel sar despeckling," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 419–422.

[14] M. V. Perera, W. G. C. Bandara, J. M. J. Valanarasu, and V. M. Patel, "Transformer-based sar image despeckling," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 751–754.

[15] E. Dalsasso, L. Denis, and F. Tupin, "Sar2sar: A semi-supervised despeckling algorithm for sar images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4321–4329, 2021.

[16] E. Dalsasso, L. Denis, and F. Tupin, "As if by magic: self-supervised training of deep despeckling networks with merlin," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[17] T. Blu and F. Luisier, "The sure-let approach to image denoising," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778–2786, 2007.

[18] C. A. Metzler, A. Mousavi, R. Heckel, and R. G. Baraniuk, "Unsupervised learning with stein's unbiased risk estimator," *arXiv preprint arXiv:1805.10531*, 2018.

[19] F. Luisier, C. Vonesch, T. Blu, and M. Unser, "Fast interscale wavelet denoising of poisson-corrupted images," *Signal processing*, vol. 90, no. 2, pp. 415–427, 2010.

[20] C. S. Seelamantula and T. Blu, "Image denoising in multiplicative noise," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 1528–1532.

[21] 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.

[22] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270–279.