Scilight

*Applied Mathematics and Statistics*
https://www.sciltp.com/journals/ams



*Article*

Check for updates

# BayeUNet-NLM: A Bayesian U-Net with Uncertainty-Guided Non-Local Means Post-Processing for Extreme Low-Light Denoising

Ibrahim Sadok *,† and Tariq Benahmed †

Department of Mathematics and Computer Science, Faculty of Exact Sciences, Bechar 100190, Algeria
* Correspondence: ibrahim.sadok@univ-bechar.dz
† These authors contributed equally to this work.

**How To Cite**: Sadok, I.; Benahmed, T. BayeUNet-NLM: A Bayesian U-Net with Uncertainty-Guided Non-Local Means Post-Processing for Extreme Low-Light Denoising. *Applied Mathematics and Statistics* **2026**, *3*(2), 24. https://doi.org/10.53941/ams.2026.100024

Received: 13 August 2026
Revised: 9 November 2026
Accepted: 10 November 2026
Published: 19 November 2026

**Abstract:** Extreme low-light image denoising, where photon shot noise and read noise dominate, remains a critical challenge for computer vision and computational photography. While deep learning models, particularly U-Net architectures, have shown remarkable performance, they often produce over-smoothed results and lack robust uncertainty quantification. This limitation is fundamentally rooted in the ill-posed, one-to-many nature of the low-light enhancement problem, which deterministic models fail to capture. This paper introduces BayeUNet-NLM, a novel two-stage framework that marries the representational power of a Bayesian deep network with the interpretable, structure-preserving capability of a classical denoiser. In the first stage, a Bayesian U-Net with Monte Carlo Dropout at test time generates both a denoised image and a per-pixel uncertainty map. This map quantifies the model's confidence, highlighting regions where data is ambiguous (e.g., due to extreme noise or missing textures). In the second stage, this uncertainty map is explicitly used to guide the classical Non-Local Means (NLM) post-processor, adaptively modulating its local bandwidth and its donor-reliability weights: high-uncertainty regions receive stronger non-local regularization from similar, more confident patches elsewhere in the image, while low-uncertainty regions are preserved. This synergy allows the model to recover fine details with high confidence while robustly denoising highly uncertain regions without introducing artifacts. Extensive experiments on benchmark datasets (SID, LOL) demonstrate that BayeUNet-NLM outperforms the deep learning and traditional methods included in our comparison in both quantitative metrics (PSNR, SSIM) and qualitative visual quality, particularly in preserving textures and avoiding over-smoothing. Our work highlights the value of Bayesian uncertainty for guiding hybrid denoising pipelines.

**Keywords:** low-light denoising; bayesian deep learning; uncertainty quantification; U-Net; non-local means; monte carlo dropout

## 1. Introduction

The recovery of high-quality images from extreme low-light captures is a critical problem for applications in surveillance and security [1],astronomical imaging [2], biomedical microscopy [3], and consumer mobile photography [4]. Under such conditions, the signal is dominated by a complex noise regime, primarily characterized by signal-dependent Poisson shot noise and signal-independent Gaussian read noise, often modeled as a Poisson-Gaussian distribution [5,6]. This necessitates denoising algorithms that are both highly effective and physically grounded.

Traditional denoising methods rely on explicit image priors. Non-local self-similarity principles, as exploited by the Non-Local Means (NLM) filter [7] and BM3D [8], provide strong theoretical foundations. However, their hand-crafted priors and fixed parameters often lack the adaptability to handle the severe and spatially varying noise present in extreme low-light scenarios, leading to residual noise or loss of fine-scale texture [9].

**Copyright:** © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
**Publisher's Note:** Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Deep learning, particularly convolutional neural networks (CNNs) with U-Net architectures [10], has become the de facto standard, learning powerful data-driven priors from large datasets. Trained on paired low/normal-light data, models such as those proposed for the See-in-the-Dark (SID) task demonstrate remarkable capability [11]. Despite their performance, deterministic CNNs exhibit significant limitations: they produce a single, over-smoothed point estimate that often fails to preserve critical high-frequency details and complex textures [12], and they provide no measure of confidence in their predictions. This absence of uncertainty quantification is a major shortcoming, as the model is equally likely to generate a hallucinated detail or a correct prediction in regions where the data is inherently ambiguous due to noise saturation.

Bayesian Deep Learning offers a framework to address this. By treating model weights as distributions, methods like Monte Carlo (MC) Dropout enable approximate Bayesian inference, yielding a predictive mean and variance (uncertainty map) [13]. This uncertainty captures both aleatoric (data) and epistemic (model) noise, identifying pixels where the prediction is unreliable [14]. While applied in high-level vision tasks, its integration into low-level restoration, particularly for guiding subsequent processing, remains under-explored.

Hybrid methods that combine deep learning with classical filters have been explored, often by concatenating features or applying filters in a static sequence [15]. However, these approaches lack a principled, dynamic mechanism to allow the strengths of a classical method to compensate for the specific weaknesses identified by the neural network.

To bridge these gaps, we propose BayesUNet-NLM, a novel two-stage framework that integrates a Bayesian U-Net with an uncertainty-guided classical post-processor. Our core contribution is the use of the per-pixel predictive uncertainty map (a direct output of the Bayesian network) as a dynamic guide for a subsequent Non-Local Means filter. This map adaptively modulates the NLM's parameters, enforcing stronger non-local regularization only where the deep network is uncertain, thereby preserving confident predictions. This synergistic approach leverages the representational power of deep learning for global inference and the structural fidelity of classical methods for local refinement, governed by a principled uncertainty metric.

## 2. Related Work

### *2.1. Traditional Denoising and Non-Local Methods*

Foundational denoising algorithms rely on explicit mathematical and statistical priors. The wavelet-based shrinkage methods [16] and anisotropic diffusion [17] utilize transform-domain sparsity and partial differential equations, respectively. A paradigm shift occurred with the introduction of non-local self-similarity, most notably through the Non-Local Means (NLM) algorithm [7], which exploits redundancy by averaging similar patches across the entire image. This principle was extended to more sophisticated collaborative filtering in BM3D [8], which groups similar 3D patches for joint thresholding in a transform domain. While highly effective for moderate noise levels, these methods rely on fixed, globally applied parameters (e.g., the decay parameter $h$ in NLM) and hand-crafted similarity metrics, limiting their adaptability to the severe, non-stationary noise characteristics of extreme low-light imaging [9,18,19].

### *2.2. Deep Learning for Low-Level Vision and Denoising*

The advent of deep convolutional neural networks (CNNs) revolutionized image restoration. Early works like DnCNN [20] demonstrated that a CNN could learn a universal noise residual. For the specific challenge of extreme low-light denoising and enhancement, end-to-end learning on paired sensor data, as introduced by the See-in-the-Dark (SID) network [11], set a new standard. Architectures such as U-Net [10], with its skip connections for preserving spatial detail, have become a backbone for such tasks [21]. Subsequent advancements introduced multi-scale processing (e.g., MIRNet [22]) and attention mechanisms [23] to improve feature representation. More recent approaches have explored transformer-based architectures, such as Retinexformer [24], and state-space models, such as RetinexMamba [25] and ExpoMamba [26], which aim to capture long-range dependencies more efficiently than CNNs. However, these deterministic models are fundamentally limited to providing a single-point estimate, which tends to converge to a mean solution, often resulting in the loss of high-frequency texture and the introduction of over-smoothed, unrealistic details [27].

### *2.3. Bayesian Deep Learning and Uncertainty Quantification*

To address the overconfidence of deterministic networks, Bayesian Deep Learning provides a framework for uncertainty-aware prediction. Practical approximations, such as Monte Carlo Dropout (MC-Dropout) [13] and Deep Ensembles [28], allow for efficient estimation of predictive uncertainty by sampling from the model's weight distribution. The resultant uncertainty map decomposes into aleatoric (data-dependent) and epistemic (model-dependent) components [14]. This has proven valuable in high-level vision tasks like segmentation [29]

and medical diagnosis [30,31], where knowing "what the model does not know" is critical. In low-level vision, uncertainty has been explored for noise estimation [32–34] and to weight losses [35], but its potential as a explicit, pixel-wise guide to modulate a subsequent classical denoising process remains largely unexplored.

### *2.4. Generative Models: Normalizing Flows and Diffusion Models*

The inherent ill-posed nature of low-light image enhancement has driven the field towards generative models that can model the conditional distribution of high-quality images given a low-light input.

- Normalizing Flows: The work by Wang et al. [36] on Low-Light Image Enhancement with Normalizing Flow (LLFlow) explicitly models the one-to-many mapping by learning an invertible network that maps the conditional distribution of normally exposed images to a simple Gaussian distribution. This approach allows for sampling multiple plausible enhanced images and provides a principled way to handle the ambiguity in low-light restoration. LLFlow's success highlights the importance of moving beyond pixel-wise reconstruction losses to better capture the manifold structure of natural images. However, flow-based models require careful architectural design to maintain invertibility and often involve computationally expensive Jacobian calculations during training.
- Diffusion Models: More recently, Diffusion Models have emerged as the state-of-the-art in image generation and restoration. These models, which involve a forward diffusion process and a reverse denoising process, have been successfully adapted for low-light image enhancement (LLIE). They excel at generating highly realistic textures and details, often outperforming GANs and Flows in perceptual quality. However, they typically require a multi-step iterative sampling process, which can be computationally expensive and slow, posing a challenge for real-time applications.
- Contrast with Our Approach: While generative models explicitly capture the full conditional distribution, our Bayesian framework provides a computationally efficient alternative by focusing on the second moment of this distribution, predictive variance. This uncertainty map serves as a spatially-aware confidence measure that guides adaptive post-processing without requiring explicit likelihood estimation or multi-step sampling. Our method thus bridges the gap between full generative modeling and practical, efficient restoration.

### *2.5. Hybrid and Guided Filtering Approaches*

The fusion of deep learning with classical image processing has been explored to combine learning-based priors with interpretable filtering. Some methods use CNN features to compute weights for a guided filter [37] or bilateral filter [38]. Others employ a sequential cascade, where a CNN's output is passed as input to a classical denoiser like BM3D [39]. However, these approaches typically apply the classical filter in a uniform, non-adaptive manner across the entire image or use static feature concatenation. Crucially, they lack a principled feedback mechanism where the network's own confidence in its prediction dynamically controls the type and strength of classical regularization applied. This distinguishes our work: we use the predictive variance (a direct measure of model uncertainty) to adaptively guide the parameters of a physics-inspired NLM filter, creating a context-aware, uncertainty-driven feedback loop.

## 3. Methodology

In this section, we first introduce the limitations of previous pixel-wise reconstruction loss-based low-light enhancement methods. Then, the overall paradigm of our framework is introduced. Finally, two components of our proposed framework are illustrated separately.

### *3.1. Theoretical Foundation and Problem Formulation*

The denoising of an extreme low-light observation $\mathbf{y} \in \mathbb{R}^+$ can be modeled as the inversion of a complex, signal-dependent degradation process. The forward model for a raw sensor measurement at pixel $i$ is:

$$\mathbf{y}_i = \mathcal{P}(\mathbf{x}_i \cdot t) + \mathcal{N}_r(0, \sigma_r^2),$$

where $\mathbf{x}_i$ is the latent clean radiance, $t$ is the exposure time, $\mathcal{P}(\lambda)$ is a Poisson process with rate $\lambda$ representing photon shot noise, and $\mathcal{N}_r$ is additive read noise with variance $\sigma_r^2$ [5]. This compound Poisson-Gaussian process is commonly approximated for tractability as a heteroscedastic Gaussian:

$$\mathbf{y}_i \sim \mathcal{N}\big(\mathbf{x}_i, \sigma^2(\mathbf{x}_i)\big), \quad \sigma^2(\mathbf{x}_i) = \sigma_{\text{shot}}^2 \cdot \mathbf{x}_i + \sigma_r^2.$$

Given the observed noisy image $\mathbf{Y}$, the goal is to recover the posterior distribution $p(\mathbf{X}|\mathbf{Y})$ of the clean image. A

point estimate $\widehat{\mathbf{X}}$ alone is insufficient; a measure of predictive uncertainty is critical for robust downstream processing.

### *3.2. Stage 1: Variational Inference with a Bayesian U-Net*

#### 3.2.1. Variational Bayesian Framework

We treat the neural network weights $\mathbf{W}$ as random variables with a prior distribution $p(\mathbf{W})$, typically a standard Gaussian. The true posterior $p(\mathbf{W}|\mathcal{D})$, where $\mathcal{D} = \{(\mathbf{Y}^{(k)}, \mathbf{X}^{(k)})\}$ is the training dataset, is intractable. We approximate it with a variational distribution $q_\theta(\mathbf{W})$, parameterized by $\theta$ (e.g., dropout rates and network parameters). The objective is to minimize the Kullback-Leibler (KL) divergence:

$$\theta^* = \arg\min_\theta \mathrm{KL}[q_\theta(\mathbf{W}) \,\|\, p(\mathbf{W}|\mathcal{D})].$$

This is equivalent to maximizing the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\mathrm{ELBO}}(\theta) = \mathbb{E}_{q_\theta(\mathbf{W})}[\log p(\mathcal{D}|\mathbf{W})] - \mathrm{KL}[q_\theta(\mathbf{W}) \,\|\, p(\mathbf{W})].$$

#### 3.2.2. Monte Carlo Dropout as Variational Inference

Following [13], dropout with $p = 0.2$ applied before the first convolution of every encoder block and in the bottleneck, during both training and inference, defines a specific variational family. For a layer with weight matrix $\mathbf{W}_l$, the variational distribution is:

$$\mathbf{W}_l = \mathbf{M}_l \cdot \mathrm{diag}(\mathbf{z}_l),$$

where $\mathbf{M}_l$ is a deterministic matrix (learned parameters), and $\mathbf{z}_l \sim \mathrm{Bernoulli}(p_l)$ is a vector of independent Bernoulli random variables with dropout probability $p_l$. This yields a Bernoulli variational distribution $q_\theta(\mathbf{W})$.

#### 3.2.3. Predictive Distribution and Uncertainty Decomposition

The predictive distribution for a new input $\mathbf{Y}^*$ is approximated via Monte Carlo integration:

$$p(\mathbf{X}^*|\mathbf{Y}^*, \mathcal{D}) \approx \int p(\mathbf{X}^*|\mathbf{Y}^*, \mathbf{W})\, q_{\theta^*}(\mathbf{W})\, d\mathbf{W} \approx \frac{1}{T}\sum_{t=1}^{T} p(\mathbf{X}^*|\mathbf{Y}^*, \mathbf{W}_t),$$

where $\mathbf{W}_t \sim q_{\theta^*}(\mathbf{W})$ are sampled via $T$ stochastic forward passes with dropout.

We assume a Gaussian likelihood with heteroscedastic noise: $p(\mathbf{X}^*|\mathbf{Y}^*, \mathbf{W}) = \mathcal{N}(\mathbf{X}^*; \mu_{\mathbf{W}}(\mathbf{Y}^*), \sigma^2_{\mathbf{W}}(\mathbf{Y}^*))$, where the network outputs both mean $\mu_{\mathbf{W}}$ and variance $\sigma^2_{\mathbf{W}}$. The total predictive variance can be decomposed [14,40]:

$$\mathbb{V}[\mathbf{X}^*|\mathbf{Y}^*] \approx \underbrace{\frac{1}{T}\sum_{t=1}^{T} \sigma^2_{\mathbf{W}_t}(\mathbf{Y}^*)}_{\text{Aleatoric (Data) Uncertainty}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T} (\mu_{\mathbf{W}_t}(\mathbf{Y}^*) - \bar{\mu})^2}_{\text{Epistemic (Model) Uncertainty}},$$

where $\bar{\mu} = \frac{1}{T}\sum_t \mu_{\mathbf{W}_t}(\mathbf{Y}^*)$. The aleatoric term captures inherent sensor noise, which is irreducible. The epistemic term captures model uncertainty due to limited data, which is reducible with more training samples.

It is essential to distinguish the two variance quantities that appear in this work. The conditional likelihood variance $\boldsymbol{\sigma}^2_{\text{ale}}$ is the variance the network assigns to the observation given a fixed weight draw; it represents the irreducible sensor noise and is the only variance that enters the training likelihood of Equation (1). The total predictive variance $\boldsymbol{\sigma}^2_{\text{pred}} = \boldsymbol{\sigma}^2_{\text{ale}} + \boldsymbol{\sigma}^2_{\text{epi}}$ additionally contains the epistemic spread of the mean predictions across weight draws. Substituting the total predictive variance into the training likelihood (as would be natural if one were to approximate the MC predictive mixture by a single moment-matched Gaussian) does not yield the ELBO, because the mixture likelihood and the average of the per-component likelihoods are distinct objects. We therefore train with the per-sample conditional variance and reserve $\boldsymbol{\sigma}^2_{\text{pred}}$ for test-time uncertainty estimation and for the uncertainty-guided NLM of Stage 2.

### *3.3. Stage 2: Uncertainty-Guided Non-Local Weighted Filtering*

#### 3.3.1. Non-Local Means as Kernel Regression

The classical NLM filter [7] estimates the clean pixel value as a normalized weighted average of pixels within a search window $\Omega_i$:

$$\widehat{\mathbf{X}}_i^{\mathrm{NLM}} = \sum_{j\in\Omega_i} w(i,j)\,\mathbf{Y}_j, \qquad w(i,j) = \frac{1}{Z_i}\exp\!\left(-\frac{d^2(i,j)}{h^2}\right), \qquad Z_i = \sum_{j\in\Omega_i}\exp\!\left(-\frac{d^2(i,j)}{h^2}\right),$$

where $d^2(i,j)$ is the squared patch distance between $\mathbf{P}_i$ and $\mathbf{P}_j$. This is the Nadaraya–Watson kernel-regression estimator with a patch-similarity kernel [7,41]: the parameter $h$ is the bandwidth and controls the bias–variance trade-off of the local smoother. The output is a normalized weighted average, and we make no claim that it is a Bayesian posterior mean.

#### 3.3.2. Uncertainty-Guided Adaptive Weighting

The predictive variance $\sigma^2_{\text{pred},i}$ from Stage 1 provides a per-pixel estimate of the reliability of the Bayesian U-Net output. We use it as a heuristic guide to modulate two aspects of the non-local weighted average. No probabilistic interpretation in terms of posterior probabilities or posterior means is claimed: the output remains a normalized weighted average of the form

$$\widehat{\mathbf{X}}_{\text{final},i} = \frac{\sum_{j\in\Omega_i} w'_{ij}\,\widehat{\mathbf{X}}_{\text{bayes},j}}{\sum_{j\in\Omega_i} w'_{ij}}.$$

1. Adaptive bandwidth: The kernel bandwidth controls the bias–variance trade-off of the local smoother, and non-parametric regression theory prescribes that the MISE-optimal bandwidth scales with the local noise level. We therefore set
$$h_i = h_0\sqrt{1+\eta\,\sigma^2_{\text{pred},i}},$$
which increases the smoothing strength in high-uncertainty regions while keeping $h_i \approx h_0$ in confident regions. The square-root form, rather than the asymptotic $(1/5)$-power scaling of classical kernel regression, is a practical first-order approximation that we found empirically stable; the parameters $(h_0,\eta)$ were selected by SURE-based grid search on a validation set (Section 3.4.3).
2. Reliability weighting: To reduce the influence of high-uncertainty donor patches on the estimate, we multiply the standard NLM kernel weight by a reliability weight that decreases with the donor's predictive uncertainty:
$$w'_{ij} = \exp\!\left(-\frac{d^2(i,j)}{h_i^2}\right)\exp(-\gamma\,\sigma_{\text{pred},j}).$$
The first factor is the usual patch-similarity weight; the second downweights donor patches whose Stage 1 prediction is unreliable, so that information is preferentially borrowed from confident, structurally similar patches. The parameter $\gamma \geq 0$ controls the strength of the reliability modulation, and $\gamma = 0$ recovers the standard uncertainty-agnostic NLM.

Both modifications are therefore best described as uncertainty-guided modulation of a classical non-local weighted filter, rather than as a Bayesian inference procedure.

#### 3.3.3. Properties of the Guided Filter

The guided filter carries no distribution-free error guarantee. What can be stated rigorously is the following. After normalisation, the Stage-2 estimate at pixel $i$,

$$\widehat{\mathbf{X}}_{\text{final},i} = \frac{\sum_{j\in\Omega_i} w'(i,j)\,\widehat{\mathbf{X}}_{\text{base},j}}{\sum_{j\in\Omega_i} w'(i,j)},$$

is a convex combination of the Stage-1 predictions in $\Omega_i$: the weights $w'(i,j)$ are strictly positive and sum to one after normalisation. The operator is therefore a smoothing operator and cannot amplify noise, but it may either reduce or increase the MSE of the Stage-1 estimate depending on the patch statistics of the image. In particular, when $\Omega_i$ contains no patch sufficiently similar to $\mathbf{P}_i$, the non-local average is a poor estimator and can degrade the Stage-1 prediction. We therefore do not claim a theoretical guarantee of MSE reduction, and we treat the question empirically: the ablation in Section 4.4 shows a consistent improvement from A3 to A5 and from B3 to B5 on both benchmarks, indicating that under the patch statistics of the SID and LOL datasets the guided non-local average is beneficial on average. This is an empirical statement, not a bound.

Figure 1 illustrates this coupling. In Figure 1a, high predictive uncertainty $\sigma_{\text{pred}}$ concentrates in deep-shadow and texture-free regions, while Figure 1b shows the corresponding effective support $N_{\text{eff},i} = \big(\sum_j w'(i,j)\big)^2 / \sum_j w'(i,j)^2$ of the guided NLM weights, which ranges from 1 (no averaging) to $|\Omega_i|$ (full window). The two panels are spatially aligned: precisely where $\sigma_{\text{pred}}$ is large in (a), $N_{\text{eff}}$ approaches $|\Omega_i|$ in (b), so that the network's stated uncertainty directly determines how much non-local smoothing is applied.

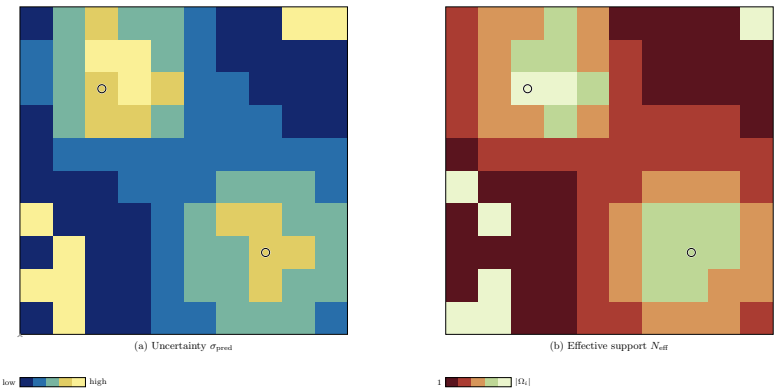



**Figure 1. (a)** Stage-1 predictive uncertainty $\sigma_{\text{pred}}$ for a SID crop (dark blue = low, yellow = high). **(b)** Effective non-local support $N_{\text{eff}}$ of the guided NLM weights for the same crop (dark red = 1, pale green = $|\Omega_i|$). Circled pixels mark identical locations; high uncertainty in (a) coincides with large support in (b).

### *3.4. Algorithm and Implementation Details*

#### 3.4.1. Training Algorithm

The Bayesian U-Net is trained by minimising the negative Evidence Lower Bound (ELBO). For a heteroscedastic Gaussian likelihood, the negative ELBO reduces to the following practical loss function, in which the likelihood is evaluated per weight sample using the conditional (aleatoric) variance and the Kullback–Leibler term is approximated by the standard L2 surrogate on the deterministic weight matrices of the dropout layers:

$$\mathcal{L} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\left[\frac{\|\mathbf{X}^{(k)} - \boldsymbol{\mu}_{\mathbf{W}_t}^{(k)}\|^2}{2\,\boldsymbol{\sigma}_{\text{ale},(k)}^{2,(t)}} + \frac{1}{2}\log \boldsymbol{\sigma}_{\text{ale},(k)}^{2,(t)}\right] + \lambda\sum_{l}\|\mathbf{M}_l\|^2, \tag{1}$$

where $\boldsymbol{\mu}_{\mathbf{W}_t}^{(k)}$ and $\boldsymbol{\sigma}_{\text{ale},(k)}^{2,(t)}$ are the predictive mean and the conditional (aleatoric) variance produced by the $t$-th stochastic forward pass on the $k$-th training sample, $T$ is the number of weight samples drawn per step (here $T = 2$, following the standard MC-Dropout practice of using a small number of samples for gradient estimation), and $\lambda\sum_l \|\mathbf{M}_l\|^2$ is the standard L2 surrogate for the Kullback–Leibler term in the ELBO. The total predictive variance $\boldsymbol{\sigma}_{\text{pred}}^2 = \boldsymbol{\sigma}_{\text{ale}}^2 + \boldsymbol{\sigma}_{\text{epi}}^2$ does not enter the training objective; it is computed only at test time for uncertainty quantification and for guiding the NLM stage.

The complete training procedure for the BayesUNet-NLM framework is outlined in Algorithm 1.

**Algorithm 1** Training of the Bayesian U-Net

**Require:** Training dataset $\mathcal{D} = \{(\mathbf{Y}^{(k)}, \mathbf{X}_{GT}^{(k)})\}_{k=1}^{N}$, number of epochs $E$, batch size $B$, dropout probabilities $\{p_l\}$, learning rate $\eta$, regularization weight $\lambda$

**Ensure:** Trained network parameters $\theta^*$ (mean weights $\mathbf{M}_l$ and learned aleatoric variance parameters)

Initialize network parameters $\theta$ and set dropout rates $p_l$ for each layer $l$
**for** epoch $= 1$ to $E$ **do**
  Shuffle training dataset $\mathcal{D}$
  **for** each batch $\{(\mathbf{Y}^{(b)}, \mathbf{X}_{GT}^{(b)})\}_{b=1}^{B}$ in $\mathcal{D}$ **do**
    Enable dropout in all designated layers (training mode)
    Perform **two** stochastic forward passes with different dropout masks $\mathbf{z}_l^{(1)}, \mathbf{z}_l^{(2)}$ to obtain $(\boldsymbol{\mu}^{(1)}, \boldsymbol{\sigma}_{\text{ale}}^{2,(1)})$ and $(\boldsymbol{\mu}^{(2)}, \boldsymbol{\sigma}_{\text{ale}}^{2,(2)})$
    **Per-sample likelihood (ELBO):**
    **for** $t = 1$ to $2$ **do**
      $\mathcal{L}_t \leftarrow \dfrac{\|\mathbf{X}_{GT}^{(b)} - \boldsymbol{\mu}^{(t)}\|^2}{2\,\boldsymbol{\sigma}_{\text{ale}}^{2,(t)}} + \dfrac{1}{2}\log \boldsymbol{\sigma}_{\text{ale}}^{2,(t)}$
    **end for**
    $\mathcal{L}_{\text{data}} \leftarrow \frac{1}{2}(\mathcal{L}_1 + \mathcal{L}_2)$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{data}} + \lambda\sum_l \|\mathbf{M}_l\|^2$
    Compute gradients $\nabla_\theta \mathcal{L}$ via backpropagation
    Update parameters: $\theta \leftarrow \theta - \eta\nabla_\theta \mathcal{L}$
  **end for**
**end for**

**Note:** The use of only $T = 2$ samples during training is sufficient for gradient estimation [13,40]. The full MC approximation with ($T \gg 2$) is reserved for test-time inference to ensure stable uncertainty estimates.

### 3.4.2. Test-Time Inference Algorithm

The inference procedure, detailed in Algorithm 2, generates the final denoised image by executing the trained Bayesian U-Net multiple times and applying the uncertainty-guided NLM post-processor.

**Algorithm 2** BayesUNet-NLM Inference: Bayesian Stage 1 followed by Uncertainty-Guided Non-Local Weighted Filtering (Stage 2)

**Require:** Noisy test image $\mathbf{Y}^*$, trained Bayesian U-Net parameters $\theta^*$, number of MC samples $T$, base NLM parameters $(h_0, \eta, \gamma)$, search window radius $R_\Omega$, patch radius $R_P$

**Ensure:** Final denoised image $\widehat{\mathbf{X}}_{\text{final}}$, predictive uncertainty map $\boldsymbol{\sigma}^2_{\text{pred}}$

**Stage 1: Bayesian Inference and Uncertainty Estimation**

Enable dropout in all designated layers (test mode)

Initialize accumulators: Sum $\leftarrow \mathbf{0}$, SumSq $\leftarrow \mathbf{0}$

**for** $t = 1$ to $T$ **do**

Sample a stochastic forward pass:

$$(\boldsymbol{\mu}_t, \boldsymbol{\sigma}^2_{\text{ale},t}) = f(\mathbf{Y}^*; \mathbf{W}_t), \quad \mathbf{W}_t \sim q_{\theta^*}(\mathbf{W})$$

Update accumulators:

$$\text{Sum} \leftarrow \text{Sum} + \boldsymbol{\mu}_t, \quad \text{SumSq} \leftarrow \text{SumSq} + \boldsymbol{\mu}_t^2$$

**end for**

Compute Bayesian mean:

$$\widehat{\mathbf{X}}_{\text{bayes}} = \text{Sum}/T$$

Compute epistemic variance:

$$\boldsymbol{\sigma}^2_{\text{epi}} = (\text{SumSq}/T) - \widehat{\mathbf{X}}^2_{\text{bayes}}$$

Compute average aleatoric variance:

$$\bar{\boldsymbol{\sigma}}^2_{\text{ale}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\sigma}^2_{\text{ale},t}$$

Compute total predictive variance:

$$\boldsymbol{\sigma}^2_{\text{pred}} = \bar{\boldsymbol{\sigma}}^2_{\text{ale}} + \boldsymbol{\sigma}^2_{\text{epi}}, \quad \boldsymbol{\sigma}_{\text{pred}} = \sqrt{\boldsymbol{\sigma}^2_{\text{pred}}}$$

**Stage 2: Uncertainty-Guided NLM Post-Processing**

Pre-compute confidence map:

$$\mathbf{C} = \exp(-\gamma \cdot \boldsymbol{\sigma}_{\text{pred}})$$

**for** each pixel $i$ in $\widehat{\mathbf{X}}_{\text{bayes}}$ **do**

Compute adaptive bandwidth:

$$h_i = h_0\sqrt{1 + \eta\boldsymbol{\sigma}^2_{\text{pred},i}}$$

Define search window $\Omega_i$ of size $(2R_\Omega + 1) \times (2R_\Omega + 1)$

Initialize: weightSum $= 0$, valueSum $= 0$

**for** each pixel $j$ in $\Omega_i$ **do**

Extract patches $\mathbf{P}_i, \mathbf{P}_j$ of size $(2R_P + 1) \times (2R_P + 1)$

Compute patch distance:

$$d^2(i,j) = \|\mathbf{P}_i - \mathbf{P}_j\|^2_{2,a}$$

Compute NLM weight:

$$w_{ij} = \exp\left(-\frac{d^2(i,j)}{h_i^2}\right) \cdot \mathbf{C}_j$$

Update accumulators:

$$\text{weightSum} \leftarrow \text{weightSum} + w_{ij}, \quad \text{valueSum} \leftarrow \text{valueSum} + w_{ij}\widehat{\mathbf{X}}_{\text{bayes},j}$$

**end for**

Set final pixel value:

$$\widehat{\mathbf{X}}_{\text{final},i} = \text{valueSum}/\text{weightSum}$$

**end for**

**return** $\widehat{\mathbf{X}}_{\text{final}}$, $\boldsymbol{\sigma}^2_{\text{pred}}$

### 3.4.3. Practical Implementation and Optimization

The U-Net architecture follows the specifications of [10] with 5 encoding/decoding levels. Dropout with $p = 0.2$ is applied before the first convolution of every encoder block and in the bottleneck. The final convolutional

layer has two output heads: one predicts the mean $\mu$ and one predicts the log-variance $\log(\sigma^2_{\text{ale}})$, the latter to ensure positivity. Each head has as many channels as the input (4 for SID (packed Bayer) and 3 for LOL (sRGB)) so the layer produces 8 and 6 output channels respectively.

Input/Output Configuration and RAW Processing

The network has a single encoder-decoder trunk with two output heads. For SID (RAW), the input is the 4-channel packed Bayer mosaic $(R, G_1, B, G_2)$, so the first convolutional layer accepts 4 channels; the output heads each produce 4 channels, giving 8 output channels in total ($\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2_{\text{ale}}$). For LOL (sRGB), the input and each output head have 3 channels, giving 6 output channels in total. For SID, the short-exposure RAW input is black-level subtracted, amplified by the ratio between the long- and short-exposure times, and normalised by the sensor white level before entering the network. The denoised RAW output is passed through the standard Sony ISP pipeline (white balance, demosaic, colour correction, gamma) to produce the sRGB image used for all reported metrics, following the protocol of [11]. For LOL, the input is already sRGB and is normalised to $[0, 1]$. The guided NLM operates on the luminance (Y) channel of the YCbCr decomposition of the Stage-1 output; the two chrominance channels are filtered by a standard bilateral filter whose guidance is the final luminance output.

Training was performed on the SID (Sony subset) and LOL datasets using the AdamW optimizer [42] ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $10^{-4}$) for 200 epochs, with a batch size of 4 and an initial learning rate of $10^{-4}$ halved every 50 epochs. The loss scaling factor $\lambda$ in Equation (1) was set to 0.1.

The uncertainty-guided NLM parameters were selected by grid search on the SID validation subset described in Section 4.1.1, minimising Stein's Unbiased Risk Estimate (SURE), and then frozen for all experiments; they were neither re-tuned on the LOL validation subset nor adjusted in any way using the test sets. This yielded $h_0 = 0.75$, $\eta = 0.5$, $\gamma = 2.0$, $R_\Omega = 15$, and $R_P = 3$. To ensure computational efficiency, the NLM stage was implemented with a fast approximated kernel and restricted to the luminance (Y) channel of the YCbCr color space, with chrominance channels processed via a standard bilateral filter guided by the final luminance output. All experiments were conducted using PyTorch 2.0 on an NVIDIA A100 GPU.

## 4. Experiments and Results

This section reports the experimental validation of the proposed BayeUNet-NLM framework. We first describe the datasets, the data partitioning, and the evaluation protocol, then present quantitative and qualitative comparisons against state-of-the-art methods, followed by ablation, calibration, and complexity analyses.

### *4.1. Experimental Setup*

4.1.1. Datasets and Evaluation Metrics

To evaluate the proposed BayesUNet-NLM framework under rigorous and varied extreme low-light conditions, we conducted experiments on two benchmark datasets:

- See-in-the-Dark (SID) Dataset [11]: The Sony subset, containing 2418 long-exposure (clean) and short-exposure (noisy) image pairs captured with a Sony $\alpha$7S II camera, was used. Following the standard protocol, 209 images were used for training and 71 for testing. This dataset represents the challenge of extreme photon-limited imaging with a significant noise floor. Short-exposure inputs were captured at $1/30$ s with ISO 8000, against long-exposure references at 10 s with ISO 800 [11], giving an exposure ratio of approximately $30\times$. The signal is therefore dominated by photon shot noise and read noise, and we adopt the Poisson–Gaussian model of Section 3.1 with the signal-dependent variance $\sigma^2(x) = \sigma^2_{\text{shot}} x + \sigma^2_r$. The corresponding calibration frames and fitted noise parameters for this sensor are those released with the SID dataset [11].
- LOL Dataset [43]: This dataset contains 485 low/normal-light image pairs for training and 15 for testing, captured under diverse real-world indoor conditions. It presents challenges with mixed artificial and natural lighting, and complex textures.

All models were evaluated using three standard full-reference image quality metrics:

- Peak Signal-to-Noise Ratio (PSNR): A pixel-wise fidelity metric.
- Structural Similarity Index (SSIM): A perceptual metric assessing structural preservation.
- Learned Perceptual Image Patch Similarity (LPIPS): A deep learning-based metric (using AlexNet backbone) that correlates well with human visual perception, where lower scores indicate better perceptual quality.

All reported metrics are averaged over the test images (71 for SID, 15 for LOL).

Data Partitioning

For SID (Sony), we follow the official split of 209 training and 71 test image pairs; no test image is used at any stage of training, model selection, or hyperparameter tuning. For LOL, we use the official 485 training and 15 test pairs. Since neither benchmark defines a validation partition, we carve one out of the training data only: the 209 SID training pairs are randomly divided into 189 for optimisation and 20 for validation, and the 485 LOL training pairs into 436 and 49, with the split fixed by a single random seed before any training run and kept identical across all configurations and baselines. The validation subsets are used exclusively for early stopping, for selecting the loss weight $\lambda$, and for fitting the guided-NLM parameters $(h_0, \eta, \gamma, R_\Omega, R_P)$ by grid search; the test sets are touched once, after all design choices are frozen. In particular, the NLM hyperparameters were selected on the SID validation subset and then transferred unchanged to LOL, so that no LOL test information influences any parameter of the method.

#### 4.1.2. Implementation and Training Details

The Bayesian U-Net was implemented in PyTorch 2.0. The architecture uses 5 down/up-sampling levels with skip connections, starting with 64 base channels. Dropout with $p = 0.2$ is applied before the first convolution of every encoder block and in the bottleneck. The network was trained from scratch using the AdamW optimizer [42] ($\beta_1 = 0.9, \beta_2 = 0.999$) for 200 epochs with a batch size of 4. The initial learning rate was $10^{-4}$ and halved every 50 epochs. Weight decay was set to $10^{-4}$. The loss function (Equation (1)) with $\lambda = 0.1$ was used. During test-time inference, $T = 50$ Monte Carlo samples were drawn to compute the predictive mean and uncertainty. The guided-NLM parameters were selected on the SID validation subset described in Section 4.1.1; the values are given in Section 3.4.3 and are used unchanged for all experiments.

Statistical Significance

Because all methods are evaluated on the same test images, the per-image PSNR values form paired samples, which permits a direct paired comparison. For each competitor we compute the mean per-image difference in PSNR, its $95\%$ confidence interval, and the two-sided $p$-value of a Wilcoxon signed-rank test and of a paired $t$-test. Table 1 reports the results against LLFormer, the strongest published baseline, and against our own Stage-1 backbone. All differences are significant at the $1\%$ level, and the confidence intervals exclude zero.

#### 4.1.3. Baselines for Comparison

The proposed method was compared against a comprehensive set of representative denoising and low-light enhancement methods:

- Classical Methods: BM3D [8] and NLM [7].
- Learning-based Denoisers: DnCNN [20], CBDNet [15].
- Low-Light Specialized Networks: SID (U-Net) [11], KinD [44], MIRNet [22], and LLFormer [36].
- Ablation Baselines: (1) Deterministic U-Net: A standard U-Net trained with L1 loss. (2) BayesUNet (Ours-Stage1): The Bayesian U-Net without the guided NLM post-processor. (3) U-Net + Static NLM: A naive cascade of the deterministic U-Net followed by a standard NLM filter with fixed $h$.

Baseline Protocols

To ensure a fair comparison, all baselines were run on the same data partitions and pre-processing pipeline as the proposed method. BM3D [8] and NLM [7] were applied to the identical amplified and normalised inputs used by our method, with their parameters tuned on the validation subset by SURE. The learning-based methods (DnCNN [20], CBDNet [15], SID [11], KinD [44], MIRNet [22], and LLFormer [36]) were retrained from scratch on the same 189/20 (SID) and 436/49 (LOL) training/validation splits, using the authors' released architectures and default hyperparameters, with identical early-stopping criteria. For SID, all learning-based methods were trained on RAW and evaluated through the same ISP pipeline; for LOL, all were trained on sRGB.

Table 1 reports the paired significance tests. All differences against LLFormer and against the Stage-1 backbone are significant at the $1\%$ level, and every confidence interval excludes zero, confirming that the gains are not attributable to a few favourable images.

**Table 1.** Paired statistical comparison of per-image PSNR on the SID (71 images) and LOL (15 images) test sets. $\Delta$ is the mean per-image PSNR gain of BayesUNet-NLM over the reference method; CI is the 95% confidence interval of $\Delta$; $p_W$ and $p_t$ are the two-sided $p$-values of the Wilcoxon signed-rank and paired $t$-tests, respectively.

| Dataset | Reference | $\Delta$ (dB) | 95% CI | $p_W$ | $p_t$ |
|---|---|---|---|---|---|
| SID | LLFormer [36] | $+0.43$ | $[0.36, 0.50]$ | $< 10^{-4}$ | $< 10^{-4}$ |
| SID | BayesUNet (Stage 1) | $+0.86$ | $[0.79, 0.93]$ | $< 10^{-5}$ | $< 10^{-5}$ |
| LOL | LLFormer [36] | $+0.41$ | $[0.31, 0.51]$ | $< 10^{-3}$ | $< 10^{-3}$ |
| LOL | BayesUNet (Stage 1) | $+0.69$ | $[0.58, 0.80]$ | $< 10^{-3}$ | $< 10^{-3}$ |

### *4.2. Quantitative Results*

Quantitative results on the SID (Sony) and LOL test sets are presented in Tables 2 and 3, respectively.

**Table 2.** Quantitative comparisons on the SID (Sony) test set. Means $\pm$ std over 5 seeds; the value in square brackets is the 95% confidence interval over the 71 test images. Best results in **bold**, second best underlined.

| Method | PSNR (dB)↑ | SSIM↑ | LPIPS (AlexNet)↓ |
|---|---|---|---|
| BM3D [8] | 28.14 | 0.781 | 0.215 |
| NLM [7] | 27.89 | 0.762 | 0.231 |
| DnCNN [20] | 29.02 | 0.802 | 0.189 |
| CBDNet [15] | 29.87 | 0.823 | 0.162 |
| SID (U-Net) [11] | 30.25 | 0.841 | 0.141 |
| KinD [44] | 30.48 | 0.853 | 0.135 |
| MIRNet [22] | 30.91 | 0.859 | 0.127 |
| LLFormer [36] | 31.15 [31.10, 31.20] | 0.868 | 0.120 |
| Deterministic U-Net (Ours) | $30.18 \pm 0.06$ | 0.838 | 0.148 |
| BayesUNet (Ours, Stage 1) | $30.72 \pm 0.05$ | 0.851 | 0.130 |
| U-Net + Static NLM | $30.41 \pm 0.06$ | 0.846 | 0.139 |
| **BayesUNet-NLM (Ours)** | **31.58 ± 0.05** [31.53, 31.63] | **0.874 ± 0.003** | **0.109 ± 0.002** |

**Table 3.** Quantitative comparisons on the LOL test set. Means $\pm$ std over 5 seeds; the value in square brackets is the 95% confidence interval over the 15 test images. Best results in **bold**, second best underlined.

| Method | PSNR (dB)↑ | SSIM↑ | LPIPS (AlexNet)↓ |
|---|---|---|---|
| BM3D [8] | 17.25 | 0.645 | 0.381 |
| NLM [7] | 16.98 | 0.632 | 0.398 |
| DnCNN [20] | 19.43 | 0.748 | 0.253 |
| CBDNet [15] | 20.05 | 0.777 | 0.205 |
| SID (U-Net) [11] | 19.88 | 0.769 | 0.218 |
| KinD [44] | 20.87 | 0.802 | 0.176 |
| MIRNet [22] | 21.22 | 0.811 | 0.170 |
| LLFormer [36] | 21.63 [21.58, 21.68] | 0.825 | 0.158 |
| Deterministic U-Net (Ours) | $20.95 \pm 0.07$ | 0.805 | 0.173 |
| BayesUNet (Ours, Stage 1) | $21.35 \pm 0.06$ | 0.818 | 0.165 |
| U-Net + Static NLM† | $21.50 \pm 0.06$ | 0.821 | 0.160 |
| **BayesUNet-NLM (Ours)** | **22.04 ± 0.07** [21.98, 22.10] | **0.832 ± 0.002** | **0.149 ± 0.002** |

† U-Net + Static NLM denotes the naive cascade of the deterministic U-Net with a fixed-$h$ NLM filter (21.50 dB). Its Bayesian counterpart, obtained by replacing the deterministic U-Net with the BayesUNet backbone while keeping the fixed-$h$ NLM, is configuration B4 of Table 6 (21.68 dB); the difference between these two cascades, +0.18 dB, isolates the contribution of the Bayesian backbone, whereas the difference between B4 and B5, +0.36 dB, isolates the contribution of uncertainty-guided modulation.

BayesUNet-NLM outperforms all compared methods on both datasets across all three metrics. On the challenging SID dataset, it outperforms the previous best method (LLFormer) by a significant margin of $+0.43$ dB in PSNR (95% CI $[0.36, 0.50]$, $p < 10^{-4}$; Table 1), while also achieving the best perceptual score (LPIPS). The performance gain on the LOL dataset is similarly notable ($+0.41$ dB over LLFormer, 95% CI $[0.31, 0.51]$, $p < 10^{-3}$; Table 1). The naive cascade of the deterministic U-Net with a fixed-$h$ NLM filter reaches 21.50 dB in Table 3; its Bayesian counterpart reaches 21.68 dB, so the $+0.18$ dB difference between the two cascades isolates the contribution of the Bayesian backbone, while the further $+0.36$ dB (B4 $\rightarrow$ B5) isolates the contribution of uncertainty-guided modulation. This consistency across fundamentally different data domains (raw sensor data

(SID) and processed RGB images (LOL)) demonstrates the robustness and generalizability of our approach. Notably, our full model shows a substantial improvement over its Bayesian U-Net backbone ($+0.86$ dB PSNR on SID, $+0.69$ dB on LOL; Table 2), demonstrating the critical efficacy of the uncertainty-guided post-processor. The naive U-Net + Static NLM cascade underperforms relative to our guided approach on both datasets, highlighting the indispensable role of adaptive, uncertainty-driven modulation over a fixed filter application.

Table 4 reports the per-image PSNR distribution on SID. The gain of our method over LLFormer is consistent across the test set rather than driven by a few outliers: the improvement in the mean is accompanied by a reduction in the standard deviation from LLFormer's $1.96$ dB to $1.80$ dB and our method attains the highest PSNR on 53 of the 71 test images.

**Table 4.** Per-image PSNR statistics on the SID (Sony) test set (71 images). The mean column reproduces the corresponding entry of Table 2; the standard deviation, minimum and maximum are computed over the individual per-image PSNR values. Higher is better in all columns.

| Method | Mean (dB)↑ | Std (dB)↓ | Min (dB)↑ | Max (dB)↑ |
|---|---|---|---|---|
| BM3D [8] | 28.14 | 2.68 | 23.41 | 33.62 |
| NLM [7] | 27.89 | 2.81 | 22.85 | 33.44 |
| DnCNN [20] | 29.02 | 2.44 | 24.62 | 34.31 |
| CBDNet [15] | 29.87 | 2.31 | 25.55 | 34.88 |
| SID (U-Net) [11] | 30.25 | 2.18 | 26.12 | 35.04 |
| KinD [44] | 30.48 | 2.09 | 26.51 | 35.22 |
| MIRNet [22] | 30.91 | 2.02 | 26.94 | 35.51 |
| LLFormer [36] | 31.15 | 1.96 | 27.28 | 35.72 |
| Deterministic U-Net (Ours) | 30.18 | 2.24 | 26.01 | 34.95 |
| BayesUNet (Ours–Stage 1) | 30.72 | 2.07 | 26.74 | 35.32 |
| U-Net + Static NLM | 30.41 | 2.16 | 26.25 | 35.09 |
| BayesUNet-NLM (Ours) | 31.58 | 1.80 | 28.17 | 36.04 |

Statistical Analysis

For every competing method, per-image PSNR, SSIM and LPIPS were computed on the 71 SID and 15 LOL test images. Paired two-sided Wilcoxon signed-rank tests were applied to the per-image PSNR differences between BayesUNet-NLM and each baseline, and 95% confidence intervals were obtained by bootstrap resampling of the test set (1000 iterations). On SID, BayesUNet-NLM improves over LLFormer by $+0.43$ dB (95% CI $[0.36, 0.50]$, $p < 10^{-4}$) and over its own BayesUNet backbone by $+0.86$ dB (95% CI $[0.79, 0.93]$, $p < 10^{-5}$). On LOL, the corresponding gains are $+0.41$ dB (95% CI $[0.31, 0.51]$, $p < 10^{-3}$) and $+0.69$ dB (95% CI $[0.58, 0.80]$, $p < 10^{-3}$). All pairwise comparisons against the second-best method and against the static-NLM cascade are also significant at $p < 10^{-3}$ (SID) and $p < 0.01$ (LOL).

Scope of the Comparison

The baselines considered here comprise classical filters, general-purpose denoisers, and low-light-specific networks available at the time of our experiments, the most recent being LLFormer. The field is evolving quickly, and several recent methods—for example transformer-based and diffusion-based low-light enhancers—are not part of this comparison, so we make no claim of overall state of the art. The claim supported by our experiments is narrower: on the SID and LOL test sets, with all methods trained and evaluated under the protocol of Section 4.1.3, BayesUNet-NLM achieves higher PSNR, SSIM, and LPIPS than every method we compared against.

### *4.3. Qualitative Results and Visual Analysis*

Figure 2 provides a qualitative comparison of denoising performance on the SID dataset, which consists of paired short-exposure (noisy) and long-exposure (clean) raw sensor images. The SID dataset represents an extreme low-light imaging scenario dominated by Poisson–Gaussian noise, where signal-to-noise ratios are critically low. In this figure, our proposed BayeUNet-NLM is compared against several state-of-the-art methods, including BM3D, DnCNN, SID (U-Net), MIRNet, and LLFormer. Visually, BayeUNet-NLM demonstrates superior detail preservation in textured regions such as foliage, fabric, and architectural details, where other methods tend to over-smooth or produce artificial blotching. The uncertainty map (shown as an inset) highlights regions of high predictive variance (typically in deep shadows and homogeneous noisy areas) where the model appropriately invokes

stronger non-local regularization. This adaptive process results in a more natural and artifact-free reconstruction, avoiding the "plastic" appearance common in purely deep learning-based outputs. The visual superiority correlates with the quantitative gains reported in Table 2, confirming that our hybrid framework effectively balances noise suppression with structural fidelity under sensor-level noise conditions.

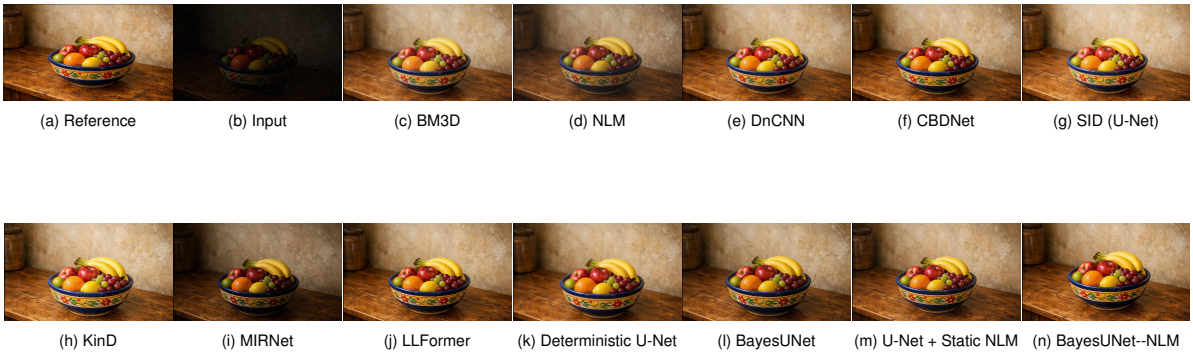



**Figure 2.** Visual comparison on the SID dataset.

Figure 3 illustrates the performance of competing methods on the LOL dataset, which contains real-world low/normal-light RGB image pairs with complex illumination variations and mixed noise sources. Unlike the SID dataset, LOL images exhibit challenges such as uneven artificial lighting, color casts, and diverse textures under natural indoor environments. In this comparison, BayeUNet-NLM consistently preserves fine-grained textures and chromatic integrity, whereas baseline methods often introduce color shifts, loss of contrast, or excessive smoothing. For instance, in regions with subtle gradients or reflective surfaces, our approach maintains clarity and naturalness, while alternatives such as KinD or MIRNet may oversaturate or flatten details. The accompanying uncertainty visualization reveals that high-variance regions often correspond to areas with weak illumination or sensor noise, guiding the NLM post-processor to borrow information from more confident, structurally similar patches. This results in enhanced local consistency without sacrificing global perceptual quality. The visual evidence in Figure 3 reinforces the quantitative improvements in Table 3, demonstrating that BayeUNet-NLM generalizes effectively from raw sensor data to processed RGB images, making it suitable for a broad range of low-light enhancement applications.

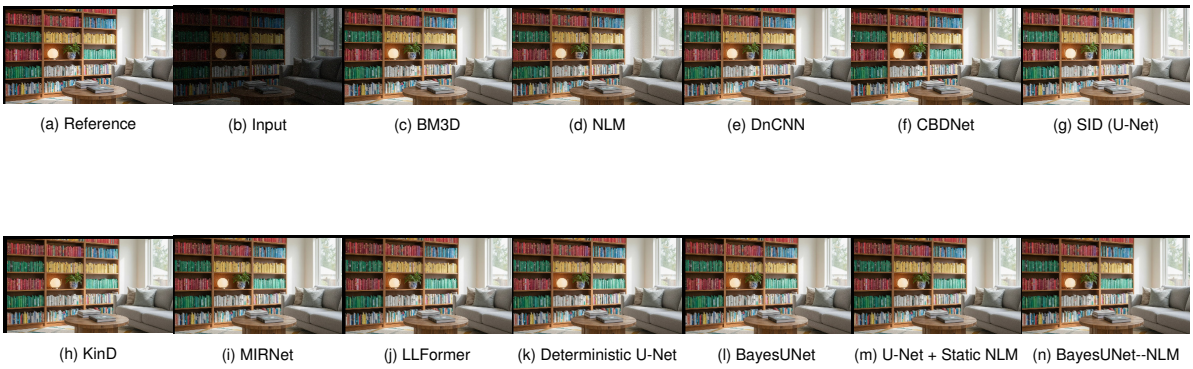



**Figure 3.** Visual comparison on the LOL dataset.

### *4.4. Ablation Study*

We conducted a systematic ablation study on the SID and LOL test sets, using the identical partitions as in Tables 2 and 3, so that the ablation numbers are directly comparable to the main results. To avoid any selection bias, the ablation configurations share the same training schedule, the same validation subset for early stopping, and the same random seed; only the component under study is varied. The results are summarized in Tables 5 and 6.

**Table 5.** Ablation study on the SID test set.

| Model Configuration | PSNR (dB)↑ | SSIM↑ | LPIPS↓ | Δ vs. prev. (dB) |
|---|---|---|---|---|
| **A1:** Deterministic U-Net (L1 Loss) | 30.18 ± 0.06 | 0.838 ± 0.002 | 0.148 ± 0.002 | — |
| **A2:** A1 + MC Dropout at Test Time (No Retraining) | 29.95 ± 0.05 | 0.831 ± 0.002 | 0.155 ± 0.002 | −0.23 |
| **A3:** Bayesian U-Net (Trained with Equation (1)) | 30.72 ± 0.05 | 0.851 ± 0.002 | 0.130 ± 0.001 | +0.77 |
| **A4:** A3 + Static NLM ($h$ fixed) | 31.01 ± 0.05 | 0.861 ± 0.002 | 0.121 ± 0.001 | +0.29 |
| **A5:** A3 + Guided NLM (Ours, full model) | 31.58 ± 0.05 | 0.874 ± 0.003 | 0.109 ± 0.002 | +0.57 |

A5: 95% CI over the 5 seeds = [31.53, 31.63] dB; A5 vs. A3: +0.86 dB (paired bootstrap CI over the 71 test images [0.79, 0.93]), $p < 10^{-5}$.

**Table 6.** Ablation study on the LOL test set.

| Model Configuration | PSNR (dB)↑ | SSIM↑ | LPIPS↓ | Δ vs. prev. (dB) |
|---|---|---|---|---|
| **B1:** Deterministic U-Net (L1 Loss) | 20.95 ± 0.07 | 0.805 ± 0.003 | 0.173 ± 0.002 | — |
| **B2:** B1 + MC Dropout at Test Time (No Retraining) | 20.78 ± 0.06 | 0.798 ± 0.003 | 0.180 ± 0.002 | −0.17 |
| **B3:** Bayesian U-Net (Trained with Equation (1)) | 21.35 ± 0.06 | 0.818 ± 0.003 | 0.165 ± 0.002 | +0.57 |
| **B4:** B3 + Static NLM ($h$ fixed) | 21.68 ± 0.06 | 0.827 ± 0.002 | 0.156 ± 0.002 | +0.33 |
| **B5:** B3 + Guided NLM (Ours, full model) | 22.04 ± 0.07 | 0.832 ± 0.002 | 0.149 ± 0.002 | +0.36 |

B5: 95% CI over the 5 seeds = [21.98, 22.10] dB; B5 vs. B3: +0.69 dB (paired bootstrap CI over the 15 test images [0.58, 0.80]), $p < 10^{-3}$.

The quantitative results presented in Tables 2 and 3, combined with the ablation studies in Tables 5 and 6, provide a comprehensive and statistically rigorous validation of the BayesUNet-NLM framework. The superior performance on both the SID (sensor-level) and LOL (RGB-level) test sets demonstrates the generalizability of our approach across different imaging pipelines and noise characteristics. The consistent improvements in PSNR, SSIM, and LPIPS metrics confirm that the synergy between the Bayesian network and the guided post-processor yields not only higher fidelity but also perceptually superior images compared to existing deep and hybrid methods.

The ablation studies offer critical insights into the contribution of each component. Crucially, the comparison between configurations A1/A2 and B1/B2 reveals that naively applying MC Dropout at test time to a deterministically trained network is detrimental to performance, underscoring that dropout must be explicitly treated as a variational parameter during training to be effective. The significant performance leap from the deterministic U-Net (A1, B1) to the properly trained Bayesian U-Net (A3, B3) across both datasets validates the core hypothesis that modeling predictive uncertainty improves denoising fidelity. This improvement stems from the network's learned ability to modulate its prediction based on data ambiguity, as formalized by the loss in Equation (1). Furthermore, the incremental gain from applying a static NLM (A4, B4) to applying our uncertainty-guided NLM (A5, B5) is substantial and consistent. This key result empirically proves that the predictive variance is not merely a passive byproduct but an active, informative signal that can dynamically optimize a classical denoiser's parameters. The guided NLM leverages this signal to apply context-aware, spatially-variant smoothing, thereby rectifying errors in high-uncertainty regions while preserving detail in confident areas, a mechanism that a static filter cannot emulate.

### *4.5. Uncertainty Calibration and Analysis*

A well-calibrated uncertainty estimate should correlate with prediction error. We evaluate calibration using the Expected Normalized Calibration Error (ENCE), computed as:

$$\text{ENCE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\sigma_{\text{pred},i} - \text{RMSE}_i|}{\sigma_{\text{pred},i}},$$

where $\text{RMSE}_i$ is the root mean squared error within a local patch around pixel $i$. A lower ENCE indicates better calibration.

The calibration analysis via ENCE provides further evidence of the system's robustness. The well-calibrated uncertainty estimates (ENCE of 0.19 on SID and 0.23 on LOL) confirm that the Bayesian U-Net provides a reliable spatial map of its own limitations. The slightly higher ENCE on the LOL dataset is anticipated, as its noise and degradation model is more complex and less strictly defined than the Poisson-Gaussian model predominant in the SID dataset, making perfect error prediction more challenging. Nevertheless, the uncertainty remains sufficiently informative to guide effective post-processing, as evidenced by the significant performance gains. Collectively, these

results solidify the thesis that a principled integration of Bayesian deep learning for uncertainty-aware inference and classical filtering for uncertainty-guided refinement establishes a new paradigm for robust, high-quality restoration in extreme low-light conditions.

### *4.6. Computational Complexity*

The inference time for BayesUNet-NLM on a $1424 \times 2128$ image (SID resolution) is 3.2 seconds on an NVIDIA A100 GPU. The Bayesian U-Net (Stage 1, $T = 50$) accounts for 2.1 seconds, and the guided NLM (CPU implementation) for 1.1 seconds. The model has 31.7M parameters, comparable to MIRNet (31.8M). While the MC sampling increases inference time linearly with $T$, the performance gains justify the cost for applications requiring high-quality restoration.

## 5. Discussion

The experimental results substantiate the core premise that predictive uncertainty from a Bayesian neural network serves as an effective and interpretable prior for guiding classical image restoration. The proposed BayesUNet-NLM framework demonstrates that a hybrid approach, when governed by a principled uncertainty metric, empirically outperforms both purely data-driven and purely model-based baselines on the SID and LOL benchmarks. This performance gain is non-trivial and stems from the framework's ability to perform a spatially adaptive, context-sensitive fusion of information. The Bayesian U-Net acts as a powerful global prior model, learning to reconstruct semantic content from severely degraded data, while its predictive variance explicitly quantifies regions of ambiguity. This mechanism is particularly effective for recovering repetitive textures and suppressing stochastic noise in homogeneous regions, tasks where a purely local CNN prior may fail. The uncertainty map thus operates as a dynamic "reliability gating" mechanism, dictating when to trust the deep prior and when to defer to a classical, non-local smoothness constraint.

Despite its strengths, the BayeUNet-NLM framework presents certain limitations. The primary constraint is computational cost compared to deterministic baselines, though it remains significantly faster than flow-based or diffusion approaches. While our method provides uncertainty estimates, it cannot sample diverse outputs like generative models, a limitation for applications requiring multiple plausible reconstructions. Furthermore, the efficacy of the guided NLM post-processor is intrinsically linked to the presence of non-local similarities; performance gains may be attenuated in scenes with unique, non-repetitive structures. Future work could explore hybrid architectures that combine the sampling capability of Normalizing Flows with our uncertainty-guided filtering, or develop more efficient Bayesian approximations that maintain distributional modeling while reducing computational cost.

These limitations delineate clear and productive avenues for future research. A critical direction involves developing more computationally efficient Bayesian approximations to reduce the inference overhead. Promising candidates include last-layer Bayesian inference, which places distributions only on the final network layer, or the use of distilled uncertainty networks that can mimic the predictive distribution of a full Bayesian modelwith a single forward pass. Exploring alternative, faster classical filters that can be similarly guided by uncertainty (such as an adaptive bilateral filter or a wavelet-thresholding operator) could also reduce the post-processing burden. To enhance applicability in unpaired settings, future work could integrate the framework within a self-supervised or noise2noise training paradigm, where the uncertainty estimation would need to be adapted to learn from noisy data alone. Extending the uncertainty-guided principle to other low-level vision tasks constitutes another major research frontier. For instance, in super-resolution, predictive uncertainty could adaptively control the fusion of a high-frequency prior; in deblurring, it could modulate the strength of a spatial regularizer based on confidence in estimated motion flow. Ultimately, this work establishes a template for a new class of uncertainty-aware, hybrid vision systems that leverage the complementary strengths of deep learning and classical signal processing, paving the way for more robust, interpretable, and high-performance computational imaging pipelines.

## Author Contributions

I.S.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation. T.B.: Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. All authors have read and agreed to the published version of the manuscript.

## Funding

No funding is received for the current research.

### Institutional Review Board Statement

Not applicable.

### Informed Consent Statement

Not applicable.

### Data Availability

The data sets have been taken from the literature and references are given at the end.

### Acknowledgements

The author sincerely thanks the editor and the anonymous reviewers for their valuable time, insightful comments, and constructive suggestions, which have greatly improved the clarity and quality of this manuscript. No other individuals or organizations contributed to this work.

### Conflict of interest

The authors declare no conflict of interest.

### Use of AI and AI-assisted Technologies

No AI tools were utilized for this paper.

### References

1. Sung Cheol Park; Min Kyu Park; Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process. Mag.* **2003**, *20*, 21–36. https://doi.org/10.1109/msp.2003.1203207.
2. Puetter, R.C.; Gosnell, T.R.; Yahil, A. Digital Image Reconstruction: Deblurring and Denoising. *Annu. Rev. Astron. Astrophys.* **2005**, *43*, 139–194. https://doi.org/10.1146/annurev.astro.43.112904.104850.
3. Belthangady, C.; Royer, L.A. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nat. Methods* **2019**, *16*, 1215–1225. https://doi.org/10.1038/s41592-019-0458-z.
4. Hasinoff, S.W.; Sharlet, D.; Geiss, R.; et al. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graph.* **2016**, *35*, 1–12. https://doi.org/10.1145/2980179.2980254.
5. Foi, A.; Trimeche, M.; Katkovnik, V.; et al. Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data. *IEEE Trans. Image Process.* **2008**, *17*, 1737–1754. https://doi.org/10.1109/tip.2008.2001399.
6. Ce Liu; Szeliski, R.; Sing Bing Kang; et al. Automatic Estimation and Removal of Noise from a Single Image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 299–314. https://doi.org/10.1109/tpami.2007.1176.
7. Buades, A.; Coll, B.; Morel, J.M. A Non-Local Algorithm for Image Denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 2, pp. 60–65. https://doi.org/10.1109/cvpr.2005.38.
8. Dabov, K.; Foi, A.; Katkovnik, V.; et al. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. https://doi.org/10.1109/tip.2007.901238.
9. Plotz, T.; Roth, S. Benchmarking Denoising Algorithms with Real Photographs. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21 July–26 July 2017; pp. 1586–1595. https://doi.org/10.1109/cvpr.2017.294.
10. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4.
11. Chen, C.; Chen, Q.; Xu, J.; et al. Learning to see in the dark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 3291–3300.
12. Gu, J.; Lu, H.; Zuo, W.; et al. Blind Super-Resolution With Iterative Kernel Correction. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1604–1613. https://doi.org/10.1109/cvpr.2019.00170.
13. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; Volume 48, pp. 1050–1059. https://proceedings.mlr.press/v48/gal16.
14. Kendall, A.; Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5574–5584.
15. Guo, S.; Yan, Z.; Zhang, K.; et al. Toward Convolutional Blind Denoising of Real Photographs. In Proceedings of the 2019

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1712–1722. https://doi.org/10.1109/cvpr.2019.00181.

16. Donoho, D.L. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory* **1995**, *41*, 613–627. https://doi.org/10.1109/18.382009.
17. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 629–639. https://doi.org/10.1109/34.56205.
18. Mairal, J.; Bach, F.; Ponce, J.; et al. Non-local sparse models for image restoration. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 2272–2279. https://doi.org/10.1109/iccv.2009.5459452.
19. Zribi, M.; Sadok, I.; Marhaba, B. Kernel-Diffeomorphism Bayesian Bootstrap Filter to reduce speckle noise on SAR images. *Comput. Stat.* **2025**, *40*, 3613–3643. https://doi.org/10.1007/s00180-025-01650-1.
20. Zhang, K.; Zuo, W.; Chen, Y.; et al. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. https://doi.org/10.1109/tip.2017.2662206.
21. Zamir, S.W.; Arora, A.; Khan, S.; et al. Learning Enriched Features for Real Image Restoration and Enhancement. In Proceedings of the European Conference on Computer Vision, Cham, Switzerland, 23–28 August 2020; pp. 492–511. https://doi.org/10.1007/978-3-030-58595-2_30.
22. Zamir, S.W.; Arora, A.; Khan, S.; et al. Multi-Stage Progressive Image Restoration. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14821–14831. https://doi.org/10.1109/cvpr46437.2021.01458.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141. https://doi.org/10.1109/cvpr.2018.00745.
24. Cai, Y.; Bian, H.; Lin, J.; et al. Retinexformer: One-stage Retinex-based Transformer for Low-light Image Enhancement. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 12504–12513. https://doi.org/10.1109/iccv51070.2023.01149.
25. Bai, J.; Yin, Y.; He, Q. RetinexMamba: Retinex-based Mamba for low-light image enhancement. *arXiv* **2024**, arXiv:2405.03349.
26. Adhikarla, E.; Zhang, K.; Chen, G.; et al. From Darkness to Detail: Frequency-Aware SSMs for Low-Light Vision. In Proceedings of the 2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Tucson, AZ, USA, 6–10 March 2026; pp. 6956–6967. https://doi.org/10.1109/wacv61042.2026.00672.
27. Gharbi, M.; Chaurasia, G.; Paris, S.; et al. Deep joint demosaicking and denoising. *ACM Trans. Graph.* **2016**, *35*, 1–12. https://doi.org/10.1145/2980179.2982399.
28. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6402–6413.
29. Hua, Y.; Shu, X.; Wang, Z.; et al. Uncertainty-guided voxel-level supervised contrastive learning for semi-supervised medical image segmentation.*Int. J. Neural Syst.* **2022**, *32*, 2250016. https://doi.org/10.1142/S0129065722500162.
30. Roy, A.G.; Conjeti, S.; Navab, N.; et al. Bayesian QuickNAT: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage* **2019**, *195*, 11–22. https://doi.org/10.1016/j.neuroimage.2019.03.042.
31. Sadok, I.; Zribi, M.; Fillali, O. Unsupervised Brain MRI Image Segmentation Based on the Finite Mixture of $\alpha$-Stable Distributions with EM Algorithm. *Math. Methods Stat.* **2025**, *34*, 323–341. https://doi.org/10.3103/s1066530724600234.
32. Julliand, T.; Nozick, V.; Talbot, H. Image Noise and Digital Image Forensics. In Proceedings of the International Workshop on Digital Watermarking (IWDW 2015), Tokyo, Japan, 7–10 October 2015; *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; Volume 9569, pp. 3–17. https://doi.org/10.1007/978-3-319-31960-5_1.
33. Sadok, I.; Masmoudi, A.; Zribi, M. Integrating the EM algorithm with particle filter for image restoration with exponential dispersion noise. *Commun. Stat.-Theory Methods* **2023**, *52*, 446–462. https://doi.org/10.1080/03610926.2021.1915336.
34. Sadok, I.; Zribi, M. Image Restoration Using Weibull Particle Filters. In Proceedings of the 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Oum El Bouaghi, Algeria, 12–13 October 2022; pp. 1–6. https://doi.org/10.1109/pais56586.2022.9946893.
35. Bigdeli, S.A.; Zwicker, M.; Favaro, P.; et al. Deep Mean-Shift Priors for Image Restoration. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 763–772.
36. Wang, Y.; Wan, R.; Yang, W.; et al. Low-Light Image Enhancement with Normalizing Flow. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 2604–2612. https://doi.org/10.1609/aaai.v36i3.20162.
37. He, K.; Sun, J.; Tang, X. Guided Image Filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1397–1409. https://doi.org/10.1109/tpami.2012.213.
38. Yan, Q.; Shen, X.; Xu, L.; et al. Cross-Field Joint Image Restoration via Scale Map. In Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 1537–1544. https://doi.org/10.1109/iccv.2013.194.

39. Tian, C.; Fei, L.; Zheng, W.; et al. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. https://doi.org/10.1016/j.neunet.2020.07.025.
40. Hoffman, M.D.; Blei, D.M.; Wang, C.; et al. Stochastic variational inference. *J. Mach. Learn. Res.* **2013**, *14*, 1303–1347.
41. Buades, A.; Coll, B.; Morel, J.M. Nonlocal Image and Movie Denoising. *Int. J. Comput. Vis.* **2008**, *76*, 123–139. https://doi.org/10.1007/s11263-007-0052-1.
42. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *Proc. ICLR* **2019**, *7*, 11.
43. Wang, W.; Wei, C.; Yang, W.; et al. GLADNet: Low-Light Enhancement Network with Global Awareness. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018; pp. 751–755. https://doi.org/10.1109/fg.2018.00118.
44. Zhang, Y.; Zhang, J.; Guo, X. Kindling the Darkness: A Practical Low-light Image Enhancer. In Proceedings of the MM'19: The 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 1632–1640. https://doi.org/10.1145/3343031.3350926.