Image Denoising: Deep Learning vs Classical Methods

Outline

- Introduction to Image Denoising
- Modeling the Image Denoising Problem
- Our attempt to deep learning-based image denoising
- Experiment 1: Gaussian Noise
- Experiment 2: Salt-and-Pepper Noise/Poisson Noise/Speckle Noise

Introduction

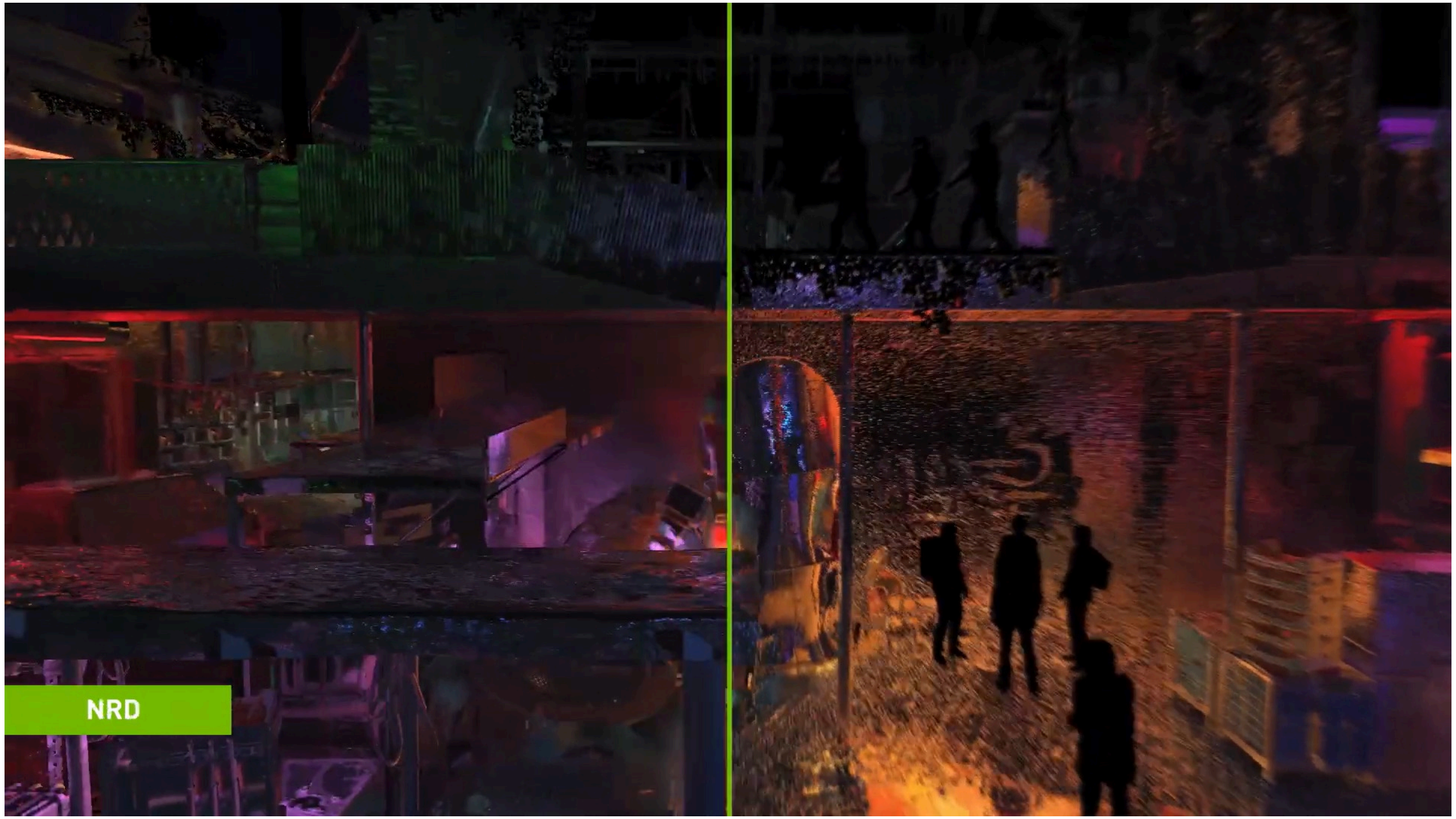
- **Image denoising** refers to the process of removing noise from an image to recover its original quality.
- Noise can result from sensor imperfections, environmental conditions, or compression artifacts.
- The goal is to enhance the visual quality by suppressing noise while preserving essential details such as edges and textures.

Motivation

- **Medical Imaging:** Crucial for accurate diagnosis and analysis.
- **Photography, Video, Game Production:** Helps achieve clearer and more aesthetically pleasing results.
- **Scientific Imaging:** Vital for extracting meaningful information from data (e.g., astronomy, microscopy).

Application: NVIDIA Real-Time Denoisers (NRD)

- **What Is Denoising?**: https://blogs.nvidia.com/blog/what-is-denoising/
- NVIDIA used image denoising to improve the visual quality of real-time ray tracing.



Challenges in Image Denoising

- Balance between noise suppression and detail preservation.
- Over-smoothing can lead to loss of important features.
- Insufficient denoising leaves residual noise.
- Considered an ill-posed problem with infinitely many plausible solutions.

History of Classical Image Denoising

Years	Core concept	Formulae for ρ(·)
~ 1970	Energy regularization	$\|x\|_2^2$
1975-1985	Spatial smoothness	$\|Lx\|_2^2$ or $\|D_v x\|_2^2 + \|D_h x\|_2^2$
1980-1985	Optimally Learned Transform	$\|Tx\|_2^2 = x^T R^{-1} x$ (via PCA)
1980-1990	Weighted smoothness	$\|Lx\|_{\mathbf{W}}^2$
1990-2000	Robust statistics	$1^T \mu(Lx)$ e.g., Hubber-Markov
1992-2005	Total-Variation	$\int_{v\in\Omega} \|\nabla x(v)\| dv = 1^T \sqrt{\|D_v x\|^2 + \|D_h x\|^2}$
1987-2005	Other PDE-based options	$\int_{v\in\Omega} g\left(\nabla x(v), \nabla^2 x(v)\right) dv$
2005-2009	Field-of-Experts	$\sum_k \lambda_k 1^T \mu_k(L_k x)$
1993-2005	Wavelet sparsity	$\|Wx\|_1$
2000-2010	Self-similarity	$\sum_{k,j\in\Omega(k)} d(R_k x, R_j x)$
2002-2012	Sparsity methods	$\|\alpha\|_0$ s.t. $x = D\alpha$
2010-2017	Low-Rank assumption	$\sum_k \|\mathbf{X}_{\Omega(k)}\|_*$

Recent developments in Deep Learning-based Image Denoising

Year	Method	Key Contributions
2021	Chen et al. (NAFNet)	Proposed NAFNet, a simplified version of channel attention for efficient image restoration, achieving state-of-the-art performance.
2022	Chen et al. (Restormer)	Introduced Restormer, focusing on computational savings with channel attention instead of spatial attention.
2022	Chen et al. (Efficient Transformers)	Developed efficient transformers with locality-constrained self-attention for image restoration.
2023	Zhang et al. (Transformers in Restoration)	Highlighted the use of transformers in low-level vision tasks like denoising, overcoming limitations of convolutional methods.
2023	Tu et al. (Fully Convolutional Methods)	Reviewed the development of fully convolutional methods in image restoration before the rise of transformers.

Mathematical Formulation for Image Denoising

Let $\mathbf{x} \in \mathbb{R}^{n \times m \times 3}$ be a clean image. We observe a noisy version of $\mathbf{x}$ given by

$$\mathbf{y} = \mathbf{x} + \mathbf{v},$$

where $\mathbf{v} \in \mathbb{R}^{n \times m \times 3}$ is a noise vector. The goal is to recover $\mathbf{x}$ from $\mathbf{y}$.

- $\mathbf{v}$ can be Gaussian noise, salt-and-pepper noise, etc.
- We will assume that the image is normalized to $[0, 1]$.
- In real-world scenarios, the noise is not known, and it may not be able to be modeled as a additive noise. However, is it often modeled as such for simplicity.

Different Types of Noise

Gaussian noise

- $\mathbf{v}$ is a Gaussian noise if $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ for some $\sigma > 0$.

Salt-and-pepper noise

- For some $p_{\max}, p_{\min} > 0$, if

$$(\mathbf{x} + \mathbf{v})_{ijc} = \begin{cases} 1 & \text{with probability } p_{\max}, \\ 0 & \text{with probability } p_{\min}, \\ x_{ijc} & \text{otherwise}, \end{cases}$$

- (c=1, 2, 3) is the color channel,

we say that $\mathbf{v}$ is a salt-and-pepper noise.

Different Types of Noise (cont'd)

Poisson noise

- $\mathbf{v}$ is a Poisson noise if $\mathbf{v}_{ijc} \sim \mathrm{Poisson}(\lambda)$ for some $\lambda > 0$.

Speckle noise

- $\mathbf{v}$ is a speckle noise if $\mathbf{v}_{ijc} \sim \mathrm{Gamma}(\alpha, \beta)$ for some $\alpha, \beta > 0$.

Remarks

- Gaussian noise is the most common type of noise that is uses in research papers.
- Most paper only consider Gaussian noise and *real-world* noise.
- These noise are used since they are easy to model and analyze.

Classical Image Denoising Methods

- In the midterm presentation, we implemented the following classical image denoising methods:
  - Mean filter: Replace each pixel with the average of its neighbors.
  - Median filter: Replace each pixel with the median of its neighbors.
  - Total variation denoising (ROF model)

$$\min_u \int_\Omega \|\nabla u\| + \frac{\lambda}{2} \int_\Omega (u - f)^2,$$

  - TV-L1 denoising

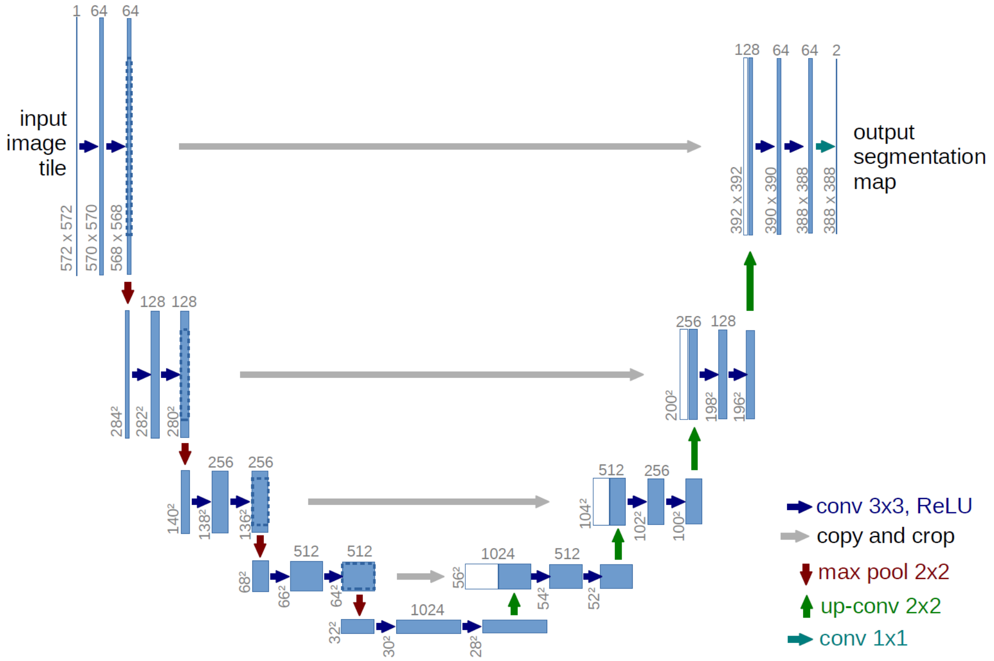
$$\min_u \int_\Omega \|\nabla u\| + \lambda \int_\Omega |u - f|,$$

Blind vs Non-blind Denoising

- Before we proceed, we need to distinguish between blind and non-blind denoising.
- In **non-blind denoising**, the noise type and level are known.
- In **blind denoising**, the noise type and level are unknown.
- Classical image denoising methods are usually blind denoising methods.
- Deep learning-based image denoising methods can be both blind and non-blind denoising methods.
- To be more specific, for deep learning-based image denoising methods, if a model is trained on a specific noise type and level, then it is a non-blind denoising method. Otherwise, it is a blind denoising method.

Our attempt to deep learning-based image denoising

- We implemented a deep learning-based image denoising method using the U-Net architecture.
- We input the noisy image and output the denoised image.
- Since the image is already normalized to $[0, 1]$, we combine the output of the U-Net with a sigmoid activation function to ensure that the output is in $[0, 1]$.



Our attempt to deep learning-based image denoising (cont'd)

- The loss function we used is the *RMSLELoss* (Root Mean Squared Logarithmic Error Loss).
- The RMSLELoss is defined as

$$\mathrm{RMSLELoss}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(1+y_i) - \log(1+\hat{y}_i))^2},$$

where $\mathbf{y}$ is the ground truth image and $\hat{\mathbf{y}}$ is the predicted image.

- The RMSLELoss is used to penalize the model more for large errors than small errors.
- To use the RMSLELoss, we need to ensure that the values are positive, hence the reason for the sigmoid activation function.

Experiment 1: Gaussian Noise

- We will consider the case where the noise is Gaussian noise.
- The goal is to predict the clean image given the noisy image with various level of Gaussian noise using a single model. This is a blind denoising problem. (Blind in the sense that the level of the noise is unknown.)
- We evaluate the model on the BSD68 dataset, with Gaussian noise level $\sigma = 15, 25, 50$. (The noise level is the standard deviation of the Gaussian noise.)
- We compare our model with other deep learning-based image denoising methods, such as NAFNet and Restormer.

Experiment 1: Gaussian Noise (cont'd)

Training setup for the U-Net model

- We train our model on the BSD400 dataset.
- In each iteration, we resize a single image to 512×512 and add Gaussian noise with 0, 15, 25, 50 standard deviation.
- We train the model for 20 epochs.
- We used the Adam optimizer with a learning rate of 10^{-4}.
- We used the *RMSLELoss* as the loss function.

Results: (Gaussian Noise)

- We compare the PSNR (Peak Signal-to-Noise Ratio) of our U-Net model with other methods.

	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
U-Net (Ours)	28.5	27.3	24.6
Mean Filter	22.6	22.4	21.8
Median Filter	23.0	22.6	22.2
ROF Model	26.1	25.9	23.7
TV-L1	26.3	25.4	23.8
Restormer	34.0	31.1	26.8
NAFNet*	22.6	13.7	4.7

- *The NAFNet pretrained model was only trained on Real-World Noise.

Comparison of NAFNet and Restormer

Model	Advantages	Disadvantages
NAFNet	Lightweight, requiring less computing resources and storage space	Lack of Self-Attention, limiting its ability to capture medium and long-range dependencies
Restormer	Use transformer to optimize Self-Attention and Feed-Forward Neural Networks for good performance	Architecture is complex and the amount of calculation is large

Image results (Gaussian Noise $\sigma = 15$)

Image results (Gaussian Noise $\sigma = 25$)

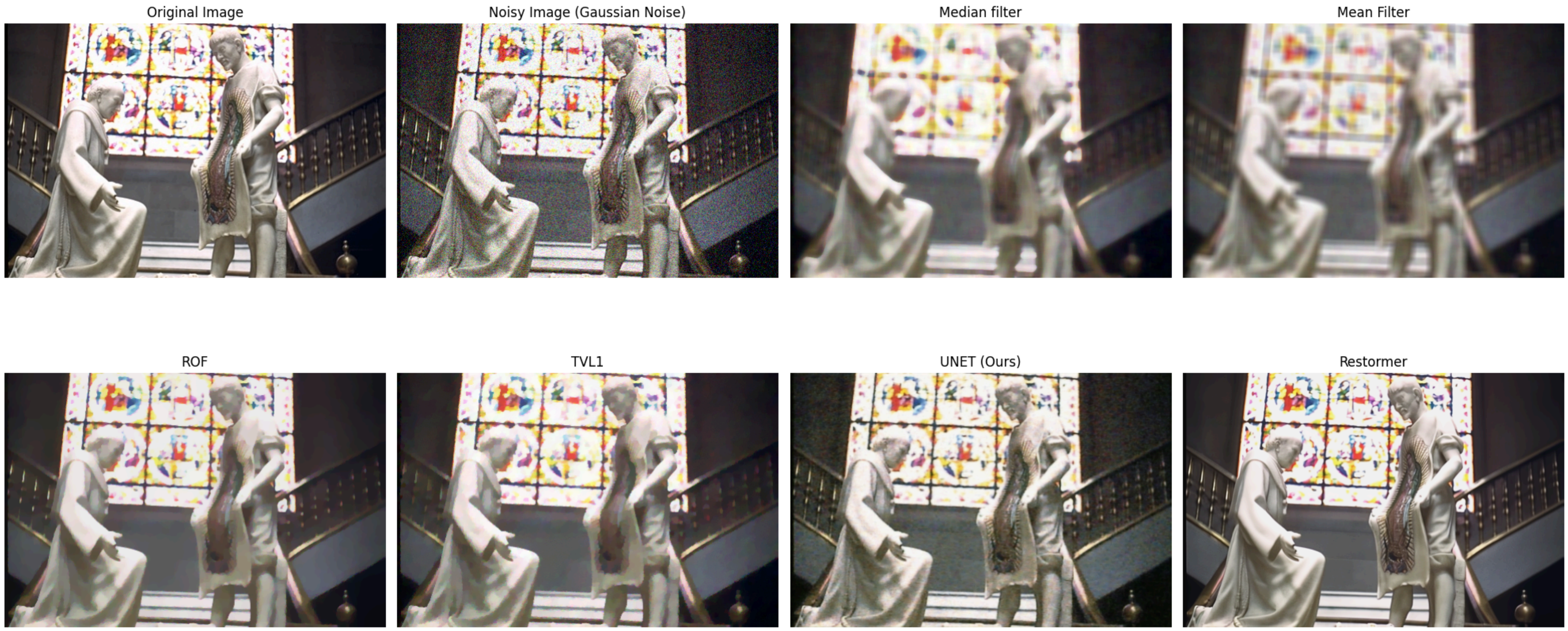


Image results (Gaussian Noise $\sigma = 50$)

Experiment 2: Salt-and-Pepper Noise/Poisson Noise/Speckle Noise

- We will consider the case where the noise is salt-and-pepper noise, Poisson noise, or speckle noise.
- The goal is to predict the clean image given the noisy image with salt-and-pepper noise, Poisson noise or speckle noise using a single model. This is a blind denoising problem. (Blind in the sense that the type of noise is unknown.)
- We construct the noisy images by adding salt-and-pepper noise, Poisson noise, and speckle noise to the BSD68 dataset. We apply a fixed noise level for each type of noise.
- We will compare our U-Net model with classical image denoising methods.

Results (Three types of noise)

- We compare the PSNR of our U-Net model with the classical methods.

	Salt-and-Pepper	Poisson	Speckle
U-Net (Ours)	26.4	29.6	26.0
U-Net (Ours)*	24.5	29.0	25.1
Mean Filter	21.7	22.6	21.8
Median Filter	23.0	23.0	22.3
ROF Model	20.6	26.2	24.2
TV-L1	27.0	26.7	24.4

- *This model was trained on Gaussian noise.

Image results (Salt-and-Pepper Noise)

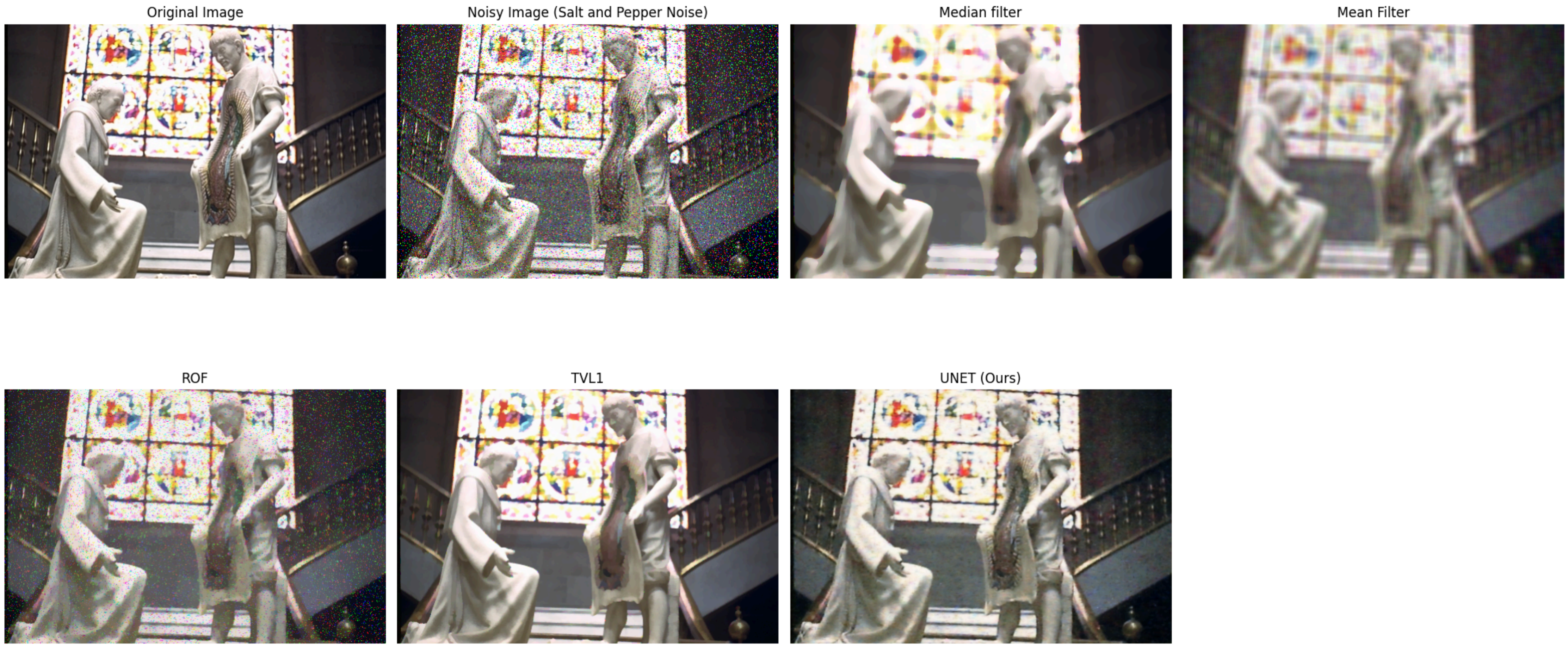


Image results (Poisson Noise)

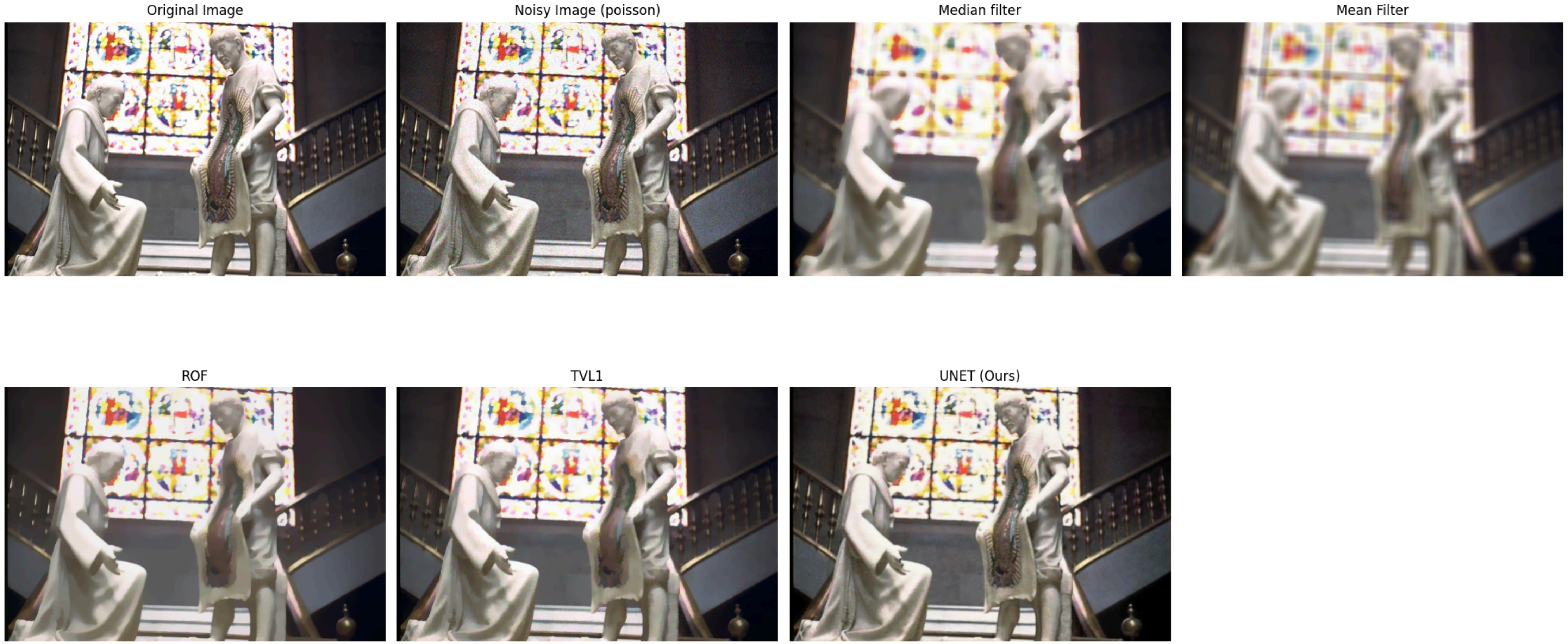
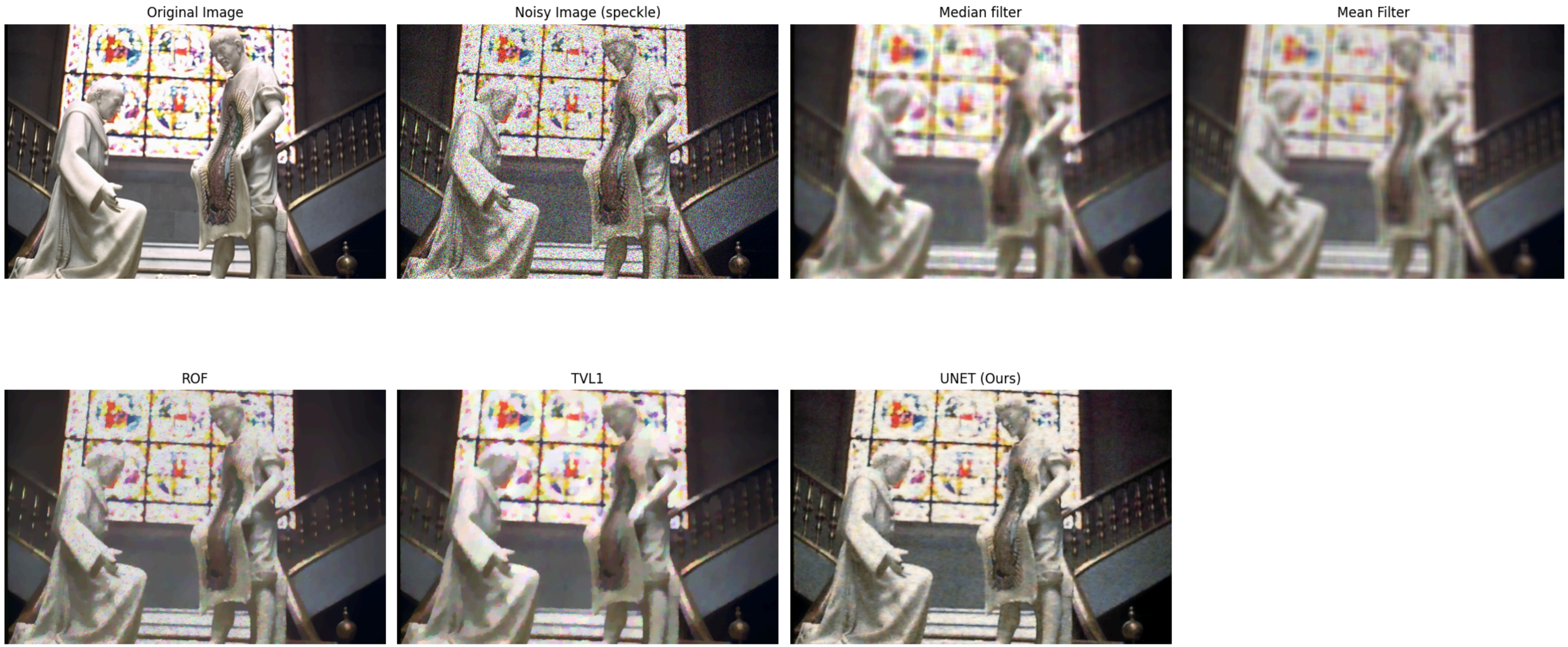


Image results (Speckle Noise)



Classical vs Deep Learning-based Image Denoising

	Advantages	Disadvantages
Classical Methods	Simple and quick to implement, Low computational requirements, Does not requrire training	Limited effectiveness on complex noise patterns, Can blur edges and fine details, Fixed method parameters often require manual tuning
Deep Learning-based Methods	High denoising performance, Better preservation of image details	High computational and data requirements, Requires large datasets for optimal training, Scalability challenges with increasing model size and complexity

Conclusion

- Image denoising is critical across various domains, and both classical and deep learning methods offer valuable solutions.
- The shift towards deep learning has shown significant improvements in denoising performance, leveraging complex architectures and large datasets.
- Continued advancements in neural network architectures and training techniques will further enhance denoising capabilities, potentially integrating more seamlessly with other image processing tasks.
- Classical methods remain relevant for their simplicity and efficiency, especially in scenarios with limited computational resources or training data.