

Image Denoising: Deep Learning vs Classical Methods

張哲嘉, 郭珈源

1 Introduction

Image denoising is a popular and fundamental problem in image processing, focusing on removing noise from an image while preserving its important features. The images we get from cameras or other devices are often corrupted by noise, which can result from various factors, such as sensor limitations, environmental conditions, or compression artifacts. This noise can degrade the visual quality of images, making image denoising crucial for enhancing both aesthetic and practical quality.

The study of image denoising is beneficial in many applications, such as medical imaging, computer vision, and photography. Effective denoising methods help suppress noise while maintaining essential details like edges and textures. In Figure 1, we show an example of an ideal denoised image. The goal of image denoising is to recover a clean image from a noisy version of it, as shown in the figure.

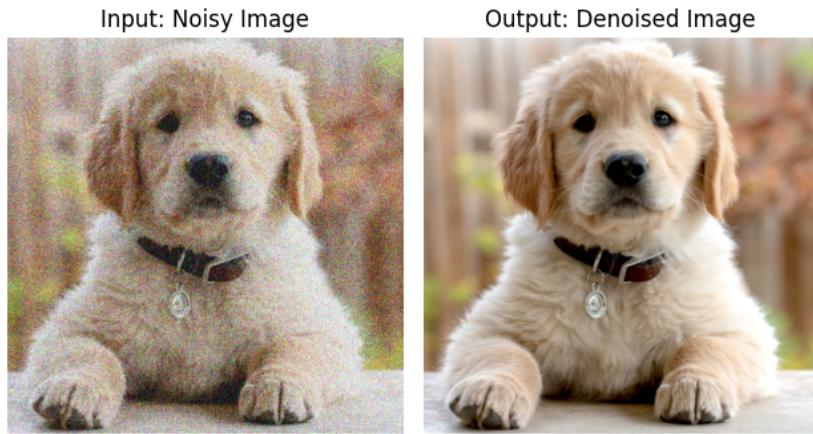


Figure 1: Ideal situation of a denoised image. Left: The input noisy image. Right: The denoised image.

1.1 Motivation

Image denoising is important in various fields. In medical imaging, it is crucial for accurate diagnosis and analysis. In photography, video, and game production, it helps achieve clearer and more aesthetically pleasing results. Additionally, in scientific imaging, it is vital to extract meaningful information from data, such as in astronomy and microscopy. Each application benefits from the enhanced clarity and detail that effective denoising provides.

1.2 Application: NVIDIA Real-Time Denoisers (NRD)

NVIDIA has used image denoising to improve the visual quality of real-time ray tracing. In the blog post "What is Desnoising?"[6], NVIDIA explains how denoising can help reduce the noise in rendered images, making them more visually appealing. NVIDIA's real-time denoisers use deep learning to remove noise from images, allowing for faster rendering times and better visual quality. Figure 2 shows an example of NVIDIA's real-time denoisers. This is an example of how deep learning-based image denoising can be used to enhance the visual quality of images in real-time applications.

1.3 Challenges in Image Denoising

The primary challenge in image denoising is balancing noise suppression with detail preservation. Over-smoothing can lead to the loss of important features, while insufficient denoising leaves residual noise. The problem is considered ill-posed because there are infinitely many plausible solutions. Achieving the right balance requires sophisticated methods that can distinguish between noise and important image features.

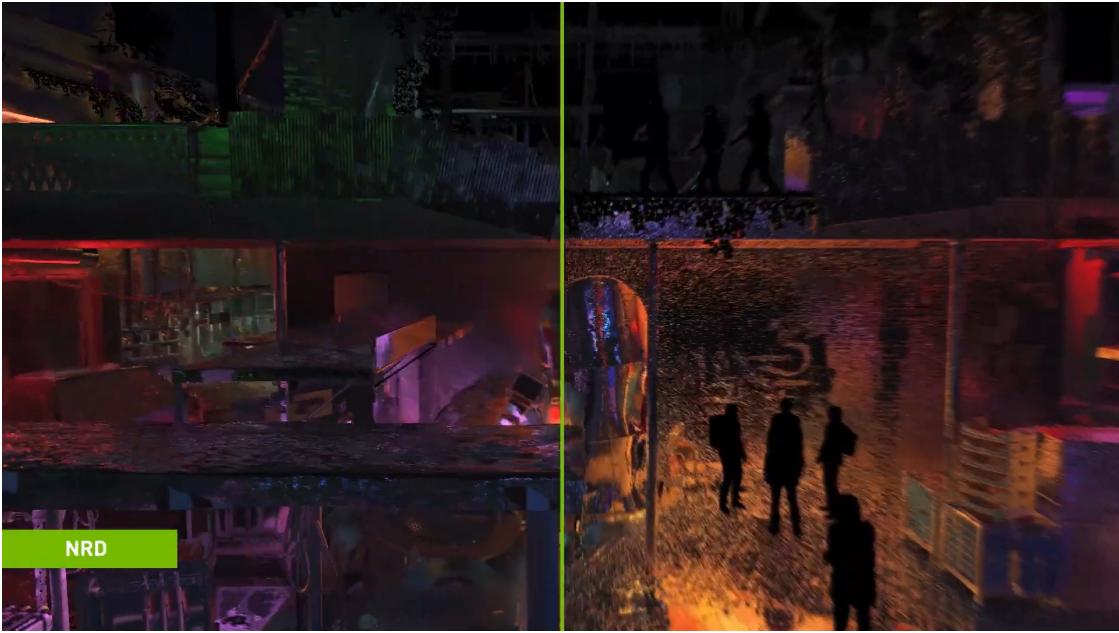


Figure 2: NVIDIA Real-Time Denoisers. The image shows the denoiser in action, removing noise from a rendered image. Image source: NVIDIA.

2 Previous Works

2.1 Classical Image Denoising

Classical image denoising has evolved significantly over the years, incorporating various techniques to improve the denoising process. Table 1 summarizes the core concepts and formulae used throughout different periods. The table is taken from [3]. In classical image denoising, we calculate the image prior $\rho(\mathbf{x})$, the probability density function of the image distribution. The prior is used to regularize the image denoising problem, balancing noise suppression and detail preservation. Generally, it is often that we minimize the following objective function:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \rho(\mathbf{x}),$$

where \mathbf{y} is the noisy image, λ is a regularization parameter, and $\rho(\mathbf{x})$ is the image prior. Different choices of $\rho(\mathbf{x})$ lead to different classical methods for image denoising. The methods listed in Table 1 are more or less based on the same principle, but they use different regularizers to model the image prior.

Years	Core Concept	Formulae for $\rho(\cdot)$
1970	Energy regularization	$\ \mathbf{x}\ _2^2$
1975-1985	Spatial smoothness	$\ \mathbf{Lx}\ _2^2$ or $\ \mathbf{D}_v x\ _2^2 + \ \mathbf{D}_h x\ _2^2$
1980-1985	Optimally Learned Transform	$\ \mathbf{Tx}\ _2^2 = \mathbf{x}^T R^{-1} \mathbf{x}$ (via PCA)
1980-1990	Weighted smoothness	$\ \mathbf{Lx}\ _{\mathbf{W}}^2$
1990-2000	Robust statistics	$\mathbf{1}^T \mu(\mathbf{Lx}')$ e.g., Hubber-Markov
1992-2005	Total-Variation	$\int_{v \in \Omega} \ \nabla \mathbf{x}(v)\ dv = \mathbf{1}^T \sqrt{\ \mathbf{D}_v x\ ^2 + \ \mathbf{D}_h x\ ^2}$
1987-2005	Other PDE-based options	$\int_{v \in \Omega} g[\nabla x(v), \nabla^2 x(v)] dv$
2005-2009	Field-of-Experts	$\sum_k \lambda_k \mathbf{1}^T \mu_k(\mathbf{L}_k \mathbf{x})$
1993-2005	Wavelet sparsity	$\ \mathbf{Wx}\ _1$
2000-2010	Self-similarity	$\sum_{k,j \in \Omega(k)} d(\mathbf{R}_k \mathbf{x}, \mathbf{R}_j \mathbf{x})$
2002-2012	Sparsity methods	$\ \alpha\ _0$ s.t. $\mathbf{x} = \mathbf{D}\alpha$
2010-2017	Low-Rank assumption	$\sum_k \ \mathbf{X}_{\Omega(k)}\ _*$

Table 1: Evolution of priors for images. The table shows the core concepts and regularizers used in classical image denoising over different periods. This table is taken from [3].

2.2 Recent Developments in Deep Learning-based Image Denoising

Recent advancements in deep learning have led to significant improvements in image-denoising techniques. Table 2 highlights some of the key contributions in this field. In the early days, fully convolutional methods were popular for image restoration. However, with the rise of transformers, researchers have developed more efficient and effective deep-learning models for image denoising. For example, Restormer and NAFNet focus on computational savings, while

Efficient Transformers use locality-constrained self-attention for image restoration. These methods have achieved state-of-the-art performance in image-denoising benchmarks. In our work, we focus on implementing a deep learning-based image-denoising method that uses a much smaller model than the state-of-the-art methods but still achieves acceptable performance.

Year	Method	Key Contributions
2022	Tu et al. [8] (Fully Convolutional Methods)	Reviewed the development of fully convolutional methods in image restoration before the rise of transformers.
2022	Zamir et al. [10] (Restormer)	Introduced Restormer, focusing on computational savings with channel attention instead of spatial attention.
2022	Chen et al. [1] (NAFNet)	Proposed NAFNet, a simplified version of channel attention for efficient image restoration, achieving state-of-the-art performance.
2022	Chen et al. [2] (Cross Aggregation Transformer)	Developed efficient transformers with locality-constrained self-attention for image restoration.
2023	Zhang et al. [11] (Attention Retractable Transformer)	Focused on combining sparse and dense attention, wherein the sparse attention module provides a wider receptive field and dense attention functions in a more local neighborhood.
2023	Xia et al. [9] (DiffIR)	An efficient diffusion model for image restoration that reduces computational costs while achieving state-of-the-art performance through a compact IR prior extraction network, a dynamic IR transformer, and a denoising network, using fewer iterations than traditional models.
2024	Ghasemabadi et al. [5] (CGNet)	Achieved state-of-the-art performance in image denoising benchmarks.

Table 2: Recent Developments in Deep Learning-based Image Denoising

3 Modeling the Image Denoising Problem

In this section, we focus on describing the mathematical formulation for image denoising. We consider a clean image \mathbf{x} and a noisy image \mathbf{y} that is a corrupted version of \mathbf{x} due to noise. The goal is to recover the clean image \mathbf{x} from the noisy image \mathbf{y} . We consider different types of noise, such as Gaussian, salt-and-pepper, Poisson, and speckle noise, and discuss how they affect the image-denoising problem.

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

3.2 Different Types of Noise

We will discuss different types of noise that are often used to model the image-denoising problem. These noise types are commonly used in research papers and are essential for understanding the image-denoising problem. Most of the noise we discuss comes from well-known probability distributions. Research papers often use these noise types to evaluate the performance of image-denoising methods.

Gaussian Noise Gaussian noise \mathbf{v} is characterized by $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ for some $\sigma > 0$, where $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ is the multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\sigma^2 \mathbf{I}$.

Salt-and-Pepper Noise For some $p_{\max}, p_{\min} > 0$, salt-and-pepper noise affects image pixels as follows:

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

where $c = 1, 2, 3$ represents the color channel.

Poisson Noise Poisson noise \mathbf{v} is characterized by $\mathbf{v}_{ijc} \sim \text{Poisson}(\lambda)$ for some $\lambda > 0$, where $\text{Poisson}(\lambda)$ is the Poisson distribution with parameter λ .

Speckle Noise Speckle noise \mathbf{v} is characterized by $\mathbf{v}_{ijc} \sim \text{Gamma}(\alpha, \beta)$ for some $\alpha, \beta > 0$, where $\text{Gamma}(\alpha, \beta)$ is the gamma distribution with shape parameter α and rate parameter β .

Remarks: Gaussian noise is the most commonly used noise type in research papers. Most papers consider Gaussian noise and real-world noise. Real-world noise is more complicated to model, but it is often modeled as additive Gaussian noise for simplicity. In Figure 3, we show the clean image and different types of noised images. We also show the absolute difference between the clean image and the noised images. This figure helps us understand how different types of noise affect the image.

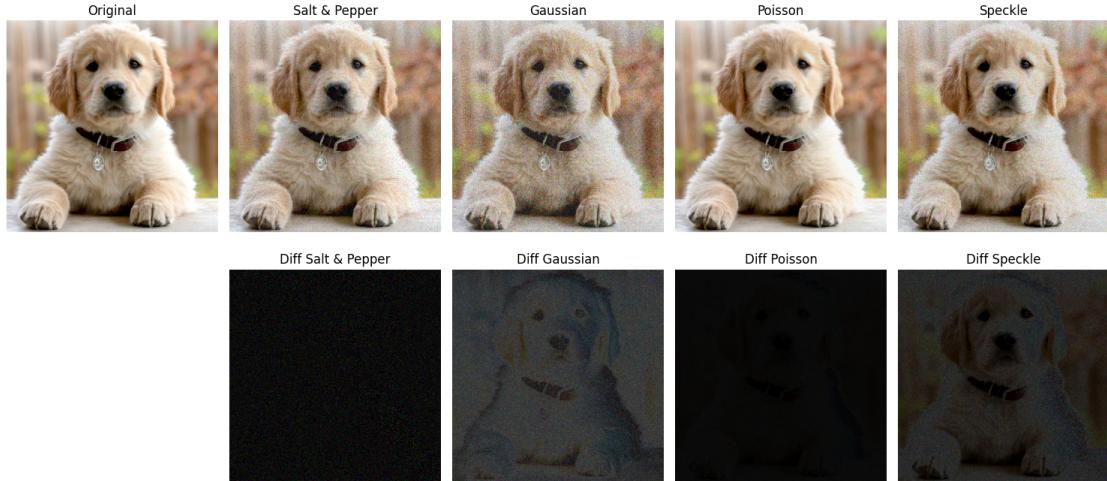


Figure 3: Upper row: clean image and different types of noised images. Lower row: the absolute difference between the clean image and the noised images.

3.3 Classical Methods

We will discuss some classical methods for image denoising. We have implemented all of the methods listed below and have experimented with different types of noise to evaluate their performance.

3.3.1 Mean filter

The mean filter can be defined as

$$\hat{x}_{ij} = \frac{1}{|W_{ij}|} \sum_{(k,l) \in W_{ij}} y_{kl},$$

where W_{ij} is a window centered at (i, j) , $|W_{ij}|$ is the number of pixels in W_{ij} , and \hat{x}_{ij} is the denoised pixel at (i, j) .

3.3.2 Median filter

The median filter is defined as

$$\hat{x}_{ij} = \text{median}\{y_{kl} : (k, l) \in W_{ij}\},$$

where W_{ij} is a window centered at (i, j) , and \hat{x}_{ij} is the denoised pixel at (i, j) .

3.3.3 Total variation denoising

The total variation method is a classical method to denoise images, which appears in [7, 4]. To explain the total variation method, it is better to introduce it in the continuous setting. Let $f : \Omega \rightarrow \mathbb{R}$ be a function, where $\Omega \subset \mathbb{R}^2$ is a bounded domain. Then f represents a noisy image we want to denoise. We will consider the ROF and TVL1 methods. Each of these methods solves an optimization problem to denoise the image.

The ROF method solves the following optimization problem:

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

where u is the denoised image, f is the noisy image, $\lambda > 0$ is a regularization parameter, and ∇u is the gradient of u . An interesting result of the ROF method is that it is equivalent to solving the Rudin-Osher-Fatemi PDE, defined as

$$\begin{cases} \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|} \right) - \lambda(u - f) = 0 & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\frac{\partial u}{\partial n}$ is the normal derivative of u .

The TVL1 method solves the following optimization problem:

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

with the same notation as above. An interesting property of the TVL1 method is that it is robust to salt-and-pepper noise.

3.4 Blind v.s. Non-blind Denoising

Before we proceed, we need to distinguish between blind and non-blind denoising. In image denoising, the noise type and level are often unknown. This leads to two types of denoising methods: blind and non-blind denoising. In non-blind denoising, the noise type and level are known, while in blind denoising, they 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, 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. In our work, we focus on blind denoising methods.

3.5 Comparision of Classical and Deep Learning-based Methods

We give a summary of the comparison between classical and deep learning-based methods in Table 3. Classical methods are based on hand-crafted features and priors, while deep learning-based methods learn features and priors from data. Deep learning-based methods have shown remarkable performance in image-denoising tasks, but they require large datasets for training and are computationally intensive. Classical methods are computationally efficient but may not achieve state-of-the-art performance.

	Advantages	Disadvantages
Classical Methods	Simple and quick to implement, Low computational requirements, Does not require 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

Table 3: Comparison of advantages and disadvantages of classical and deep learning-based denoising methods.

4 Our Attempt at Deep Learning-based Image Denoising

In this section, we describe our attempt at deep learning-based image denoising. We have implemented a deep learning-based method for image denoising and experimented with different types of noise to evaluate its performance. Our method is based on a U-Net architecture, which is a popular choice for image restoration tasks. We combine the U-Net architecture with a sigmoid activation function to model the image-denoising problem. The use of a sigmoid activation function allows our model to output pixel values in the range $[0, 1]$, which is suitable for image-denoising tasks. We also used a special loss function: the root mean squared logarithmic error loss (RMSLELoss) to train our model. The RMSLELoss is a variant of the root mean squared error (RMSE) loss that reduces the impact of large errors in the loss calculation. The model architecture is summarized in Figure 4. We will give a bit more details of each component in the following sections.

4.1 U-Net Architecture

The U-Net architecture is a popular choice for image restoration tasks due to its ability to capture both local and global features. The U-Net architecture consists of an encoder-decoder structure with skip connections that help preserve spatial information. For our implementation, we used a U-Net architecture with 5 down-sampling and up-sampling blocks. Each

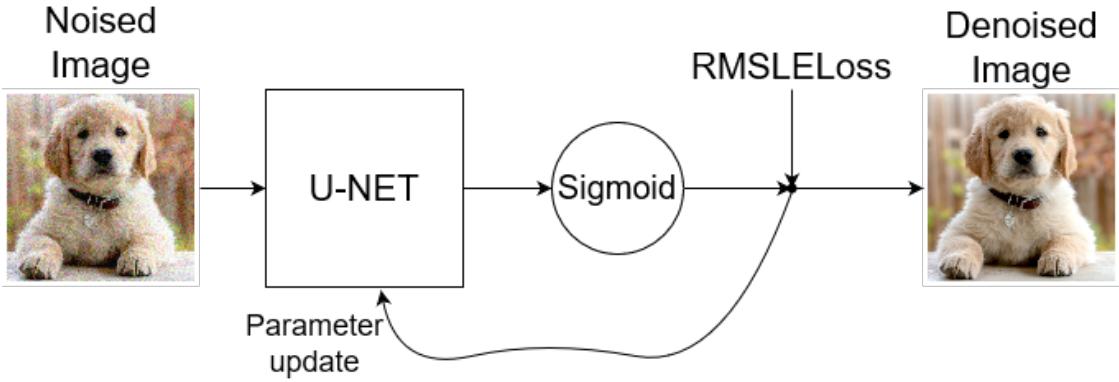


Figure 4: Model architecture for deep learning-based image denoising. The model is based on a U-Net architecture with a sigmoid activation function. The model is trained using the RMSLELoss loss function.

block consists of two convolutional layers followed by a batch normalization layer and a ReLU activation function. Skip connections are added between the encoder and decoder blocks to combine low-level and high-level features. The size of the convolutional filters is 3×3 , and the number of filters is doubled at each down-sampling block and halved at each up-sampling block. 64, 128, 256, 512, and 1024 filters are used in the down-sampling blocks, while 512, 256, 128, and 64 filters are used in the up-sampling blocks. The final output of the U-Net architecture is a denoised image with the same dimensions as the input noisy image, which is height n , width m , and 3 color channels.

4.2 Sigmoid Activation Function

The sigmoid activation function is used in our model to output pixel values in the range $[0, 1]$, which is suitable for image-denoising tasks. The sigmoid activation function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

where x is the input to the activation function. The sigmoid function maps the input to the range $[0, 1]$, making it suitable for modeling pixel values in images. The use of the sigmoid activation function ensures that the output of our model is within the valid pixel value range.

4.3 RMSLELoss Loss Function

The root mean squared logarithmic error loss (RMSLELoss) is a variant of the root mean squared error (RMSE) loss that reduces the impact of large errors in the loss calculation. The RMSLELoss is defined as

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

where y_i is the ground truth pixel value, \hat{y}_i is the predicted pixel value, and n is the number of pixels in the image. Although logically thinking, there aren't really large outliers in the pixel values; the RMSLELoss is still used in our model to train the model. We think that the RMSLELoss can help the model focus on preserving the details in the image while suppressing the noise. In Figure 5, we show the comparison between absolute error loss and RMSLELoss.

5 Experiments

We will do two experiments to evaluate the performance of our deep learning-based image denoising method. We will compare the performance of our method with classical methods for image denoising and some deep learning-based methods. In the experiments, we use the pixel signal-to-noise ratio (PSNR) as the evaluation metric. The PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{1^2}{\text{MSE}} \right),$$

where MSE is the mean squared error between the clean image and the denoised image, a higher PSNR value indicates better denoising performance. Our model is trained on the BSD400 dataset, which contains 400 images. We add different noise sources for different experiments. The test set is the BSD68 dataset, which contains 68 images.

5.1 Experimental 1: Gaussian Noise with Different Levels of Noise

In this experiment, we add Gaussian noise with different levels of noise to the clean images in the BSD68 dataset. We experimented with the ability of our model to denoise images when the noise level was different. During the training

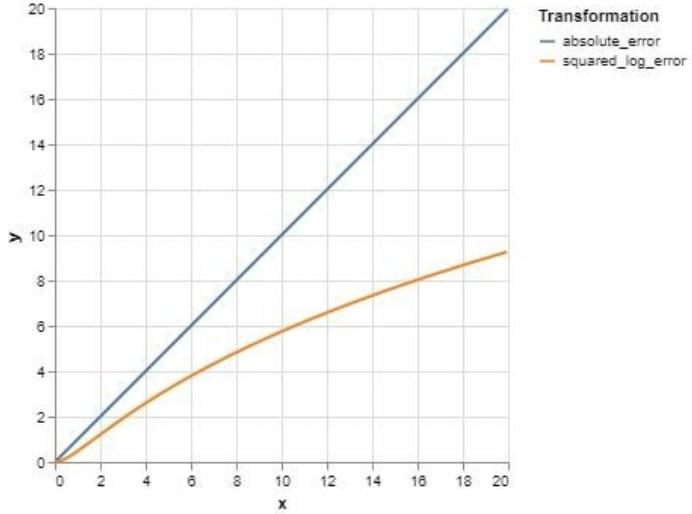


Figure 5: Comparison between absolute error loss and RMSLELoss. The RMSLELoss reduces the impact of large errors in the loss calculation. Source: <https://builtin.com/data-science/msle-vs-mse>.

process, in each iteration, we train the model with a batch containing an image with 0, 15, 25, and 50 standard deviations of Gaussian noise. Note that the standard deviation of the Gaussian noise is normalized to [0, 1] by dividing 255. We will evaluate the performance of our model on the test set, which contains images with different levels of Gaussian noise. The result is shown in Table 4. In Figure 6, Figure 7, and Figure 8, we show the comparison of different methods for experiment 1. The σ is set to 15, 25, and 50, respectively. As we can see from the table and figures, our deep learning-based method outperforms the classical methods. The performance is not as good as the state-of-the-art methods, but it is acceptable given the simplicity of our model. While our model is able to denoise images with different levels of Gaussian noise, at a higher noise level, the performance of our model decreases. We can also see that the color shifts towards a more blueish color as the noise level increases.

	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
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 [10]	34.0	31.1	26.8
NAFNet [1]*	22.6	13.7	4.7

Table 4: Comparison of different methods at varying noise levels (σ). *The NAFNet is only trained on real-world noise but not Gaussian noise, so the performance is not as good as that of the other methods.

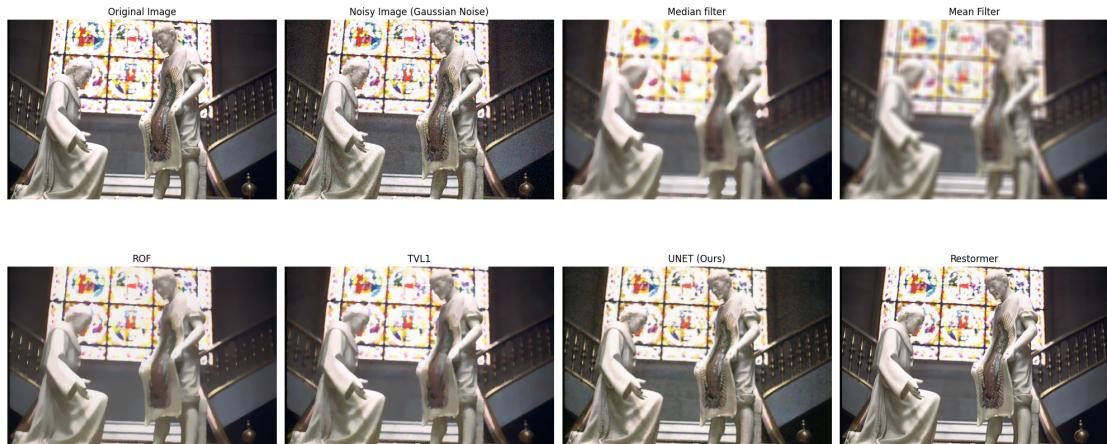


Figure 6: Comparison of different methods for experiment 1. The σ is set to 15.

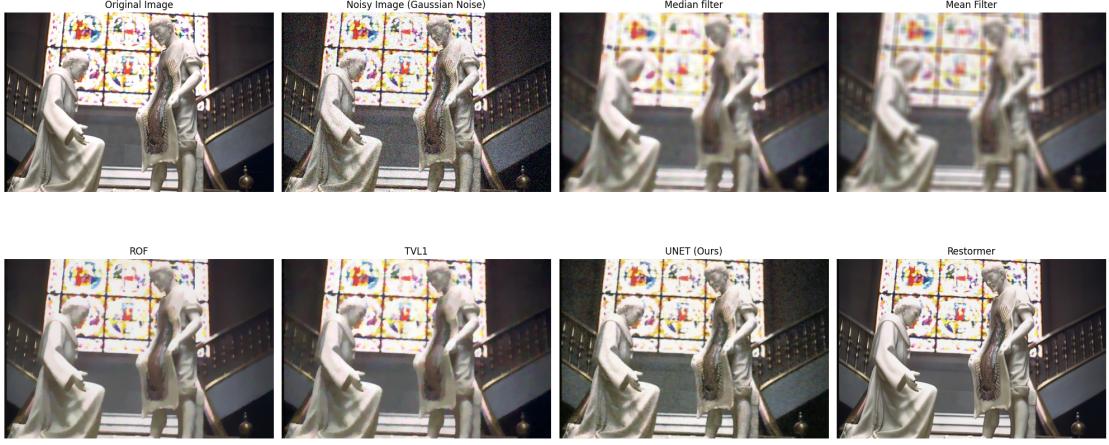


Figure 7: Comparison of different methods for experiment 1. The σ is set to 25.



Figure 8: Comparison of different methods for experiment 1. The σ is set to 50.

5.2 Experimental 2: Salt-and-Pepper/Poisson/Speckle Noise

In this experiment, we add salt-and-pepper, Poisson, and speckle noise to the clean images in the BSD68 dataset. We experimented with the ability of our model to denoise images when the noise type was different. During the training process, in each iteration, we train the model with a batch containing an image of salt-and-pepper, Poisson, and speckle noise. We will evaluate the performance of our model on the test set, which contains images with different types of noise. The result is shown in Table 5. In Figure 9, Figure 10, and Figure 11, we show the comparison of different methods for experiment 2. As we can see from the table and figures, our deep learning-based method outperforms 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

Table 5: Comparison of different methods under various noise types. *: The U-Net model is trained on Gaussian noise, but it is still able to denoise images with different types of noise.

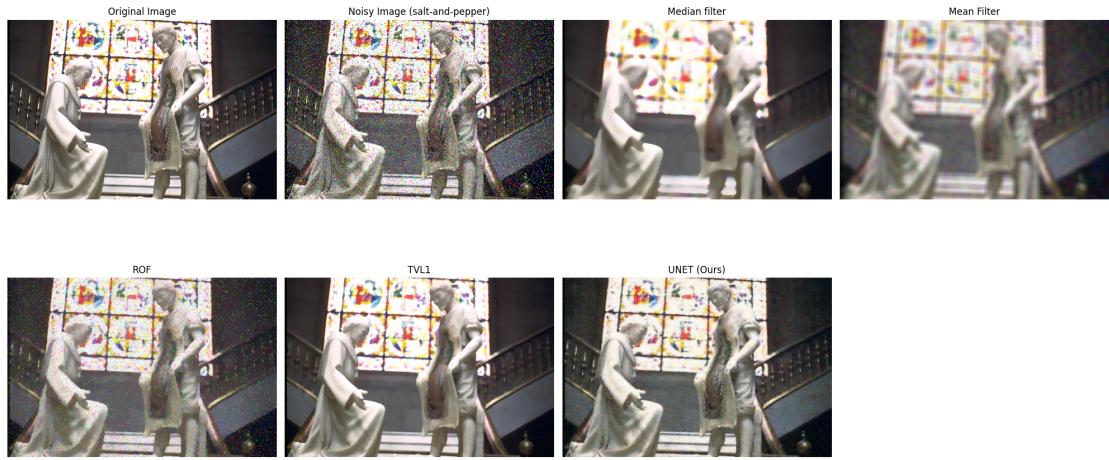


Figure 9: Comparison of different methods for experiment 2. The noise type is salt-and-pepper noise.

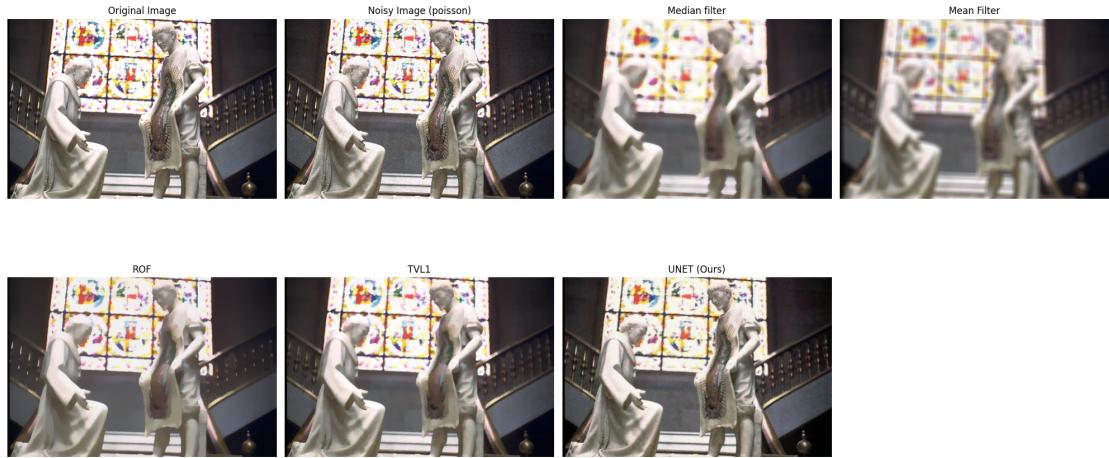


Figure 10: Comparison of different methods for experiment 2. The noise type is Poisson noise.

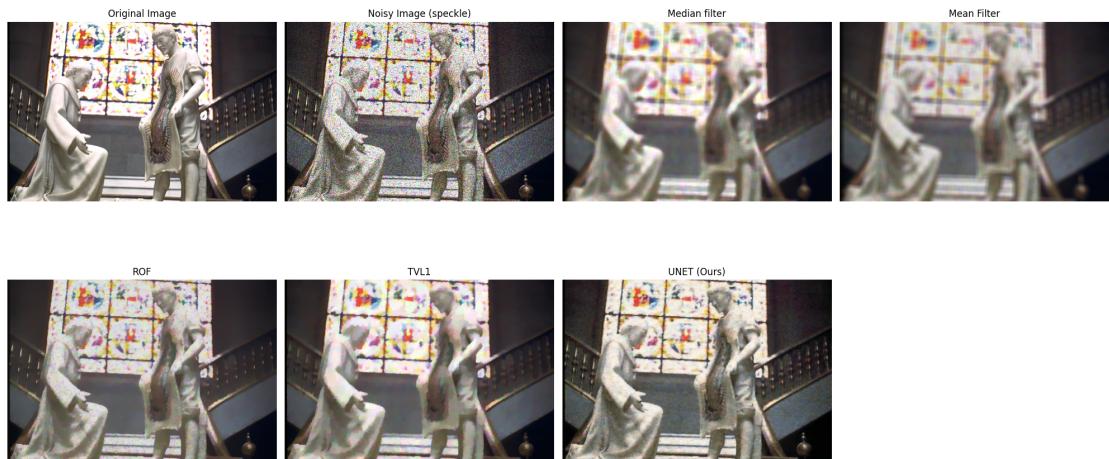


Figure 11: Comparison of different methods for experiment 2. The noise type is speckle noise.

6 Conclusion

Image denoising is a challenging problem due to the trade-off between noise suppression and detail preservation. Classical methods use different regularizers to model this problem. In recent years, deep learning-based methods have shown remarkable performance in image denoising. However, these methods require large datasets for training and are computationally intensive. Future research may focus on developing efficient and effective deep learning-based methods for image denoising. Most of the recent deep learning-based methods use transformers and very large models. Thus, we wanted to explore the possibility of using a smaller model for image denoising. Although we did not perform as well as the current deep learning-based methods, our model still achieved acceptable performance, beating the classical methods. However, classical methods are still useful when the training data is limited or when computational resources are scarce. In conclusion, image denoising is an essential problem in image processing, and both classical and deep learning-based methods play a crucial role in addressing this problem. We hope that we can further improve our model towards more efficient and effective image denoising in the future.

References

- [1] Liangyu Chen et al. “Simple baselines for image restoration”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 17–33.
- [2] Zheng Chen et al. “Cross Aggregation Transformer for Image Restoration”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 25478–25490.
- [3] Michael Elad, Bahjat Kawar, and Gregory Vaksman. “Image Denoising: The Deep Learning Revolution and Beyond —A Survey Paper”. In: *SIAM Journal on Imaging Sciences* 16.3 (2023), pp. 1594–1654. DOI: 10.1137/23M1545859. eprint: <https://doi.org/10.1137/23M1545859>. URL: <https://doi.org/10.1137/23M1545859>.
- [4] Pascal Getreuer. “Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman”. In: *Image Processing On Line* 2 (2012). <https://doi.org/10.5201/ipol.2012.g-tvd>, pp. 74–95.
- [5] Amirhosein Ghasemabadi et al. “CascadedGaze: Efficiency in Global Context Extraction for Image Restoration”. In: *Transactions on Machine Learning Research* (2024). ISSN: 2835-8856. URL: <https://openreview.net/forum?id=C3FXHxMVuq>.
- [6] NVIDIA. *What Is Denoising?* <https://blogs.nvidia.com/blog/what-is-denoising/>. 2022.
- [7] Leonid Rudin, Stanley Osher, and Emad Fatemi. “Nonlinear total variation based noise removal algorithms”. In: *Physica D: Nonlinear Phenomena* 60 (Nov. 1992), pp. 259–268. DOI: 10.1016/0167-2789(92)90242-F.
- [8] Zhengzhong Tu et al. “Maxim: Multi-axis mlp for image processing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5769–5780.
- [9] Bin Xia et al. “Diffir: Efficient diffusion model for image restoration”. In: *arXiv preprint arXiv:2303.09472* (2023).
- [10] Syed Waqas Zamir et al. “Restormer: Efficient transformer for high-resolution image restoration”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5728–5739.
- [11] Jiale Zhang et al. “Accurate Image Restoration with Attention Retractable Transformer”. In: *The Eleventh International Conference on Learning Representations*. 2022.