Evaluation of Denoising Diffusion Probabilistic Models in Denoising Tasks

Xuduo Gu, and Miaoqi Zhang

Abstract—Image denoising has been a popular topic in the field of image processing, and efforts have been made to produce higher-quality denoised images. With more generative models invented in the past decades, it could also be possible to denoise images using denoising models. This paper experiments with the denoising diffusion probabilistic model (DDPM) [1] in image denoising. The performance of the models is measured by the quality of the denoised image and will be compared with other models. PSNR, PSNR-HVS-M (PHM) [2], and structural similarity (SSIM) [3] are used as metrics to measure the quality of the denoised images. We also found a mapping from the noise levels to the step number of the denoising stage to produce the best result. To further explore the performance of these models, we tested DDPM on images with different levels of Gaussian and Poison noises and the ones that are out of the distribution of the training set.

Index Terms—Computational Photography, Denoising, Generative Models, Denoising Diffusion Probabilistic Models

1 Introduction

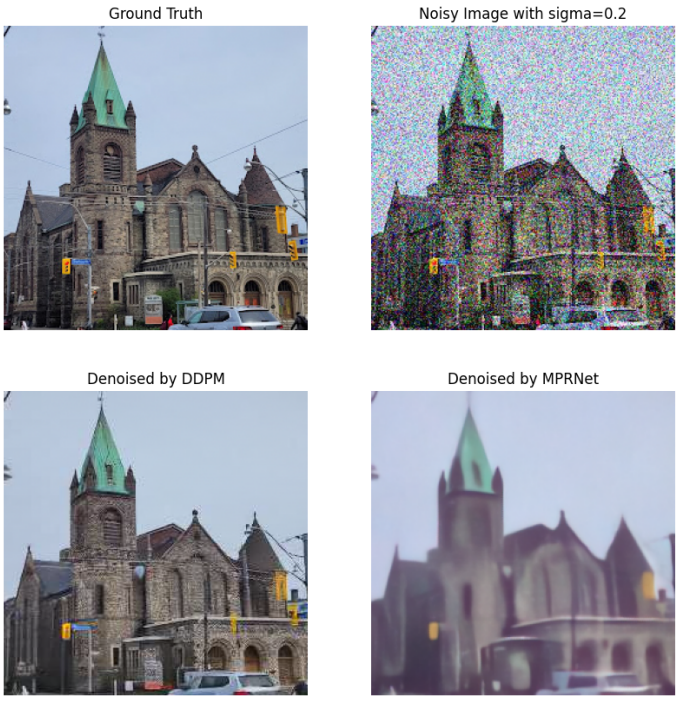
Noises in images are annoying, and that makes denoising a popular topic for image processing. The importance and popularity of denoising could be reflected in the diversity of denoising models. These models propose methods from very different perspectives. Some models use varied types of filters to erase the noises, while some other models adopt a deep learning strategy and use neural networks to achieve the same goal. Aside from the models that process the original images, there are also models that generate images from scratch to approximate the denoised image (e.g., the deep image prior [4]). If a model generates images by removing noises, then how well does this generative model perform on denoising?

The denoising diffusion probabilistic model (DDPM) [1], [5] is a generative model that creates high-resolution images through denoising. DDPM is trained by continuously adding Gaussian-distributed noises to images until the images become isotropic-like noises and then learning to find the priors of the noisy images. When pure Gaussian-distributed noises are passed to a trained DDPM model, the model will continuously take denoising steps and finally generate a clear image. The intermediate result of each denoising step will just be an image with different levels of Gaussian noises [1].

Since DDPM could eventually generate a clear result from these noisy intermediate images, it is intuitive to believe that DDPM will also perform denoising well on the noisy images injected into the middle of the denoising stage [6]. This work adopts such a method and explores the performance of DDPM in removing different levels of noise from images. The performance is measured using PSNR, PSNR-HVS-M (PHM) [2], and structural similarity (SSIM).

We show that DDPM could generate better results than bilateral filters [7] and the multi-stage progressive image restoration model (MPRNet) [8] on in-distribution images. Results also indicate that Poisson noises added on top of Gaussian noises could interfere with and lower the performance of DDPM. Meanwhile, we also show that DDPM's good performance in this scenario cannot be generalized to out-of-distribution images when the noise level is high.

Fig. 1: Noisy church image (retrieved from Google Map) denoised using pretrained DDPM model [9] and MPRNet [8].



- *Xuduo Gu is with the Department of Computer Science, University of Toronto, Toronto, ON, M5S 1A4. E-mail: xdgu@cs.toronto.edu*
- *Miaoqi Zhang is with the Department of Computer Science, University of Toronto, Toronto, ON, M5S 1A4. E-mail: miaoqi@cs.toronto.edu*

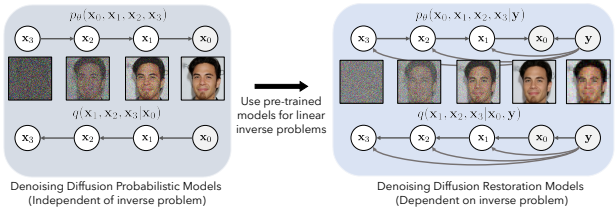
2 Related Work

Classical models mitigate noises by consulting neighbouring areas or similar areas of the image to estimate the real value of the pixels, such as the bilateral filter [7] and the non-local means filter [10]. While these filters are efficient in denoising, the processed image could suffer from blurry artifacts as the filters discourage sharp edges.

As deep learning went viral, neural networks are also applied to the topic of denoising. For instance, Zhang et al. proposed DnCNN model [11], which uses a neural network and batch normalization to fit the noises and subtract them from the noisy image. Still, DnCNN might also leave noticeable artifacts in the image. This problem could somehow be mitigated by the deep image prior model as it could achieve super-resolution for the image and therefore keep the image details [4]. However, the deep image prior model relies on an early stopping for the Unet fitting process, which means a deviated estimation of the number of iterations could lead to either removing image details or leaving noises in the image.

The denoising diffusion restoration models (DDRM) [6] adopt DDPM as the denoising tool in image restoration and could generate outstanding outputs for general linear inverse problems. This work uses DDPM in a similar way as DDRM and tests how well DDPM denoises images without relying on other features of DDRM (e.g., Memory Efficient SVD). While DDRM is claimed to be able to handle out-of-distribution images if it utilizes a DDPM model trained on a multi-modal dataset, it remains unclear if DDPM could perform as well if it were trained on a single class of images. Meanwhile, since DDPM is trained with Gaussian noises, previous works did not evaluate the model when the image is also interfered with by Poisson noises. We test the performance of DDPM in these scenarios to provide more evaluations on the generalizability of DDPM in denoising.

Fig. 2: DDRM uses a pretrained DDPM model. Figure by [6]



3 Experiment

To accommodate the limited computing resources available for this work, we use the pretrained DDPM model [9] that is also used by DDRM. This model was previously trained on the LSUN church [12] datasets. It takes in an image array $\hat{\mathbf{x}}$, a current step t at which the denoising process starts, and the total number of denoising steps n_s. Each step of denoising will decrement t by 1, and the denoising process will stop if t reaches 0 or all the n_s steps are executed. Since the choices of parameters could significantly influence the quality of the results, it is crucial to find the appropriate parameters to generate the most desired output.

In order to find the optimal starting steps to denoise images of different noise levels, we will first calculate the closed form of the mapping from the noise variance to the optimal starting step. To verify this mapping, we fine-tune the optimal t value for each noise level and use the interpolation of these results to match the theoretical mapping. We prepare pairs of clean images $\mathbf{x}$ and noisy images $\tilde{\mathbf{x}}$. We obtained a small batch of images of churches from Google Maps and cropped them to the size of 256 $\times$ 256 pixels. Gaussian noises with $\sigma = 0.05, 0.1, 0.2, 0.4$ are then added to these images to generate noisy images. For each level of noise, we automatically tune the model by finding

$$\arg\max_{t,n_s} \sum_{i=1}^{n} \text{PSNR-HVS-M}(\mathbf{x}_i, \text{DDPM}(t, n_s, \tilde{\mathbf{x}}_i))$$

where n is the number of clean images. After parameters are tuned according to PSNR-HVS-M, we manually fine-tune the model around the previous results and find the final choice of parameters based on both PSNR-HVS-M and SSIM [3]. PSNR is not used in tuning as it is found to be biased toward blurry images, which might cause the tuning process to overestimate the step number.

After generating the denoised results using the DDPM with the optimal parameters, we compare these images with the ones that are processed by a fine-tuned bilateral filter and a pretrained MPRNet [8] in terms of PSNR, PSNR-HVS-M, and SSIM. The results of this comparison could offer insight into whether DDPM performs denoising on Gaussian noises as satisfactory as other models that are designed for denoising. To evaluate if DDPM could denoise more complex noises, we also use DDPM to denoise images with Gaussian-Poisson noises. We also use DDPM to denoise images of bridges and owls with Gaussian noises and check if DDPM could properly denoise out-of-distribution images when it is trained on a single class of images.

4 Experiment Result and Analysis

4.1 Optimal Denoising Step Number

When the DDPM model is trained, the variance of the noise added to the image is linear to the diffusion step number t. The variance is expressed as

$$\beta_t = \frac{0.0199}{999} * (t-1)) + 0.0001$$

for the pretrained model we use [1] [9]. Considering that each diffusion step also dilutes the image from the previous step by $\prod_{s=1}^{t} 1 - \beta_s$, the noises of all the steps are not accumulated by simple addition. The noise variance at each diffusion step could be calculated using the function [1] [9]

$$f(t) = \sigma_t^2 = 1 - \prod_{s=1}^{t}(1-\beta_s)$$

Since denoising for DDPM is the reverse process of diffusion, we expect the noise variance to be the same for the same step t of diffusion and denoising. The same function f also maps the denoising step number to the corresponding noise variance. Therefore, the inverse of this function, namely f^{-1}, gives the step number for each noise variance by

$$t = f^{-1}(\sigma_t^2)$$

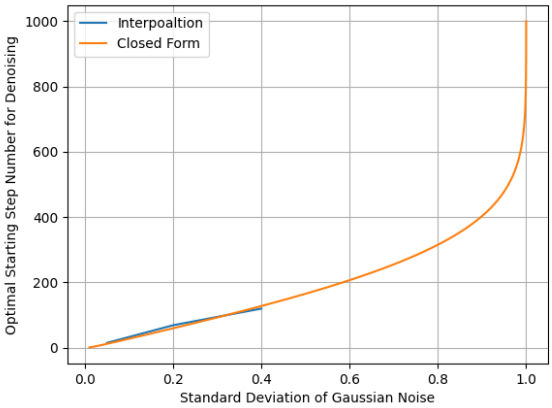
After the theoretical analysis, we also adopted the empirical strategy to tune the parameters. Table 1 describes the tuning result based on the previously described strategy. It confirms our intuition that optimal t is positively correlated to the noise level and the resulting quality peaks when n_s is the largest possible value.

TABLE 1: Optimal t and n_s for denoising Gaussian noises.

-	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
t	15	33	69	120
n_s	15	33	69	120

The interpolation based on the tuned parameters verifies the previously calculated function f^{-1}. Figure 3 shows that when σ is within the range of $(0.05, 0.4)$, the curve of the interpolation is almost overlapped with the curve of f^{-1}. Notice that we only tuned the model in the range of $(0.05, 0.4)$ as this work is regarding image denoising instead of image generation, and higher noise levels could cover the semantics of the image. Meanwhile, as the noisy images are clipped so that the pixel values fit in the range of (0, 1), the actual noise variance of the noisy image will be significantly lower than the amount added to the clean image when the noise variance is very high.

Fig. 3: Comparison of Theoretical Mapping and Interpolation

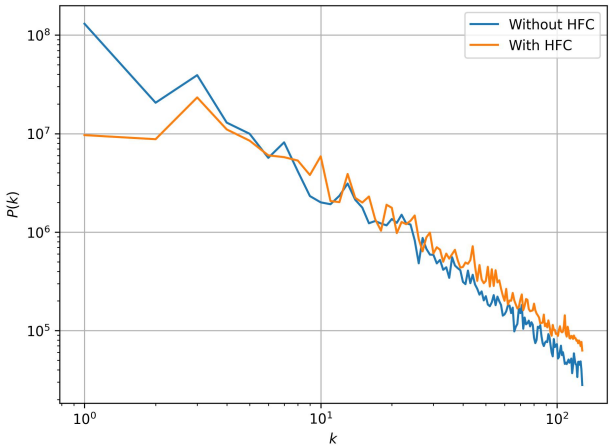


4.2 Quantitative Analysis

With the optimal parameters, DDPM could generate denoised results with very high SSIM and decent PSNR-HVS-M with respect to the ground truth (Table 2 and Table 3).

In order to better evaluate if DDPM is performing denoising as well as other denoising models, we compared DDPM with a fine-tuned bilateral filter and MPRNet on two batches of in-distribution noisy images (adapted on images retrieved from Google Maps) that are not seen by DDPM during training or tuning. The first batch of images is generated by adding Gaussian noises to images with high-frequency areas (e.g., trees and bricks), while the second batch is created from images with lower frequency (e.g., white walls and sky). The frequencies of the images are also measured in terms of the power spectrum. To calculate the power spectrum, we first converted the image into the Fourier domain using discrete Fourier Transformation, then we calculated the power of the amplitude of each frequency. The power spectrum shows that the images in the first batch have higher power at high frequencies (Figure 4).

Fig. 4: Power spectrum of the images with and without high-frequency component (HFC).



When the noise level is very low, DDPM is slightly outperformed by bilateral filters. However, when the noise level is not low ($\sigma \geq 0.1$), DDPM significantly outperforms both bilateral filters and MPRNet on denoising images with high-frequency components in terms of both PSNR-HVS-M and SSIM (Table 2). That said, the images denoised by MPRNet have PSNR-HVS-M values that are even lower than the noisy input, which indicates that the images denoised by MPRNet could be undesired in certain sense. The case is different for low-frequency images. While these noisy images have lower PSNR-HVS-M and SSIM values than their noisy counterparts with higher frequencies, they end up with higher measurements after being denoised by DDPM or MPRNet. Although DDPM still generates better results than MPRNet in terms of PSNR-HVS-M, the difference in the SSIM values for each noise level is insignificant.

4.3 Qualitative Analysis

When the noise level is low (e.g., $\sigma \leq 0.1$), DDPM could generate denoised images that are visually very close to the original images. This is also the case for MPRNet. Table 4 and Table 5 show denoised results of images with and without high-frequency components. As the variance of Gaussian noise increases, the DDPM-denoised images remain realistic while the ones processed by MPRNet gradually become blurry. Compared to MPRNet, DDPM generally preserves more details of the image during denoising despite the high levels of noise. One example is that when $\sigma = 0.4$, the windows are still in the image denoised by DDPM, and they are not detectable if denoised by MPRNet (Table 6).

However, DDPM could still miss details when the noise level is high enough to interfere with the original information encoded in the image. For example, the blossom tree in the middle of the images in Table 4 has a frequency that is very close to the Nyquist frequency due to the

TABLE 2: Comparison of bilateral filter (BF), MPRNet, and DDPM in denoising images with high-frequency areas.

-	$\sigma = 0.05$			$\sigma = 0.1$			$\sigma = 0.2$			$\sigma = 0.4$		
Metrics	PSNR	PHM	SSIM	PSNR	PHM	SSIM	PSNR	PHM	SSIM	PSNR	PHM	SSIM
Noisy	28.53	33.18	0.88	22.67	26.68	0.66	16.68	20.10	0.37	12.27	14.48	0.19
BF	**29.82**	**34.46**	0.93	26.84	28.16	0.86	22.23	21.98	0.70	16.01	16.67	0.32
MPRNet	29.26	31.19	0.93	24.46	23.72	0.86	20.03	17.83	0.76	16.29	13.15	0.65
DDPM	28.84	34.34	**0.93**	**26.95**	**28.89**	**0.90**	**24.23**	**23.43**	**0.84**	**20.26**	**17.49**	**0.76**

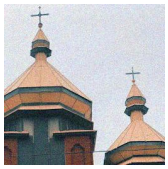
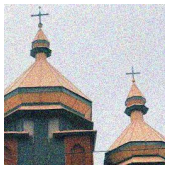
TABLE 3: Comparison of bilateral filter (BF), MPRNet, and DDPM in denoising images without high-frequency area.

-	$\sigma = 0.05$			$\sigma = 0.1$			$\sigma = 0.2$			$\sigma = 0.4$		
Metrics	PSNR	PHM	SSIM	PSNR	PHM	SSIM	PSNR	PHM	SSIM	PSNR	PHM	SSIM
Noisy	29.25	32.36	0.85	22.73	26.27	0.57	16.69	19.97	0.29	12.26	14.36	0.14
BF	**31.83**	34.60	0.94	28.44	28.79	0.87	23.48	23.33	0.71	16.19	17.06	0.27
MPRNet	31.44	32.80	**0.95**	26.55	25.50	0.92	21.23	18.79	**0.85**	16.90	13.89	0.77
DDPM	31.51	**35.66**	0.94	**29.51**	**30.96**	**0.92**	**25.59**	**25.51**	0.83	**21.08**	**18.46**	**0.80**

TABLE 4: Noisy images with high-frequency component denoised by DDPM and MPRNet

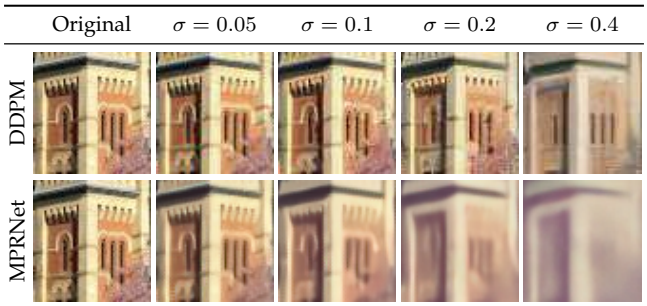
Original	-	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
	Noisy				
	DDPM				
	MPRNet				

TABLE 5: Noisy images denoised by DDPM and MPRNet (only low-frequency)

Original	-	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
	Noisy				
	DDPM				
	MPRNet				

low resolution of the image. DDPM fails to preserve the semantics of this component and just treats it as noises when the noise standard deviation is $\sigma = 0.4$, so DDPM processes this area as a part of the church and removes the blossom tree. Since low-frequency images generally do not suffer from noises interfering with the high-frequency component, DDPM could preserve almost all the semantics even with high noise levels as illustrated in Table 5.

TABLE 6: Denoised high-frequency component (32× 32 pixels).



4.4 Generalization to Other Types of Noise

DDPM is trained with Gaussian noises for both diffusion and denoising steps, which makes it perform denoising well on images with only Gaussian noises. In real-life situations such as digital photography, it is often the case that images do not only contain Gaussian noises but also fall victim to Poisson noises [13]. To evaluate how well DDPM denoises images with Gaussian-Poisson noises, we first added Poisson noises with the same variance as Gaussian noises to the images such that the Poisson noises will not be overwhelmed by the Gaussian noises. Then, we used DDPM to denoise these images with the step number t computed according to the given Gaussian noise variance and the function f^{-1}.

Results show that the Poisson noises do influence DDPM's ability to denoising. The denoising process does not increase the PSNR-HVS-M and SSIM values of the images with Gaussian-Poisson noises as much as the images with only Gaussian noises (see Table 8 in Appendix). When the noise variance is low, DDPM could still leave Poisson-like noises in the denoised results. As the variance increases, these Poisson-like noises become less significant, but there will still be noticeable yet undesired dotty artifacts in the low-frequency area of the denoised results.

We also compared DDPM with bilateral filters and MPRNet on a more realistic noise model with which noises are added to the image to mimic the noises in a real raw image [14]. Results (see Table 9 in Appendix) imply that DDPM does not demonstrate an obvious advantage over the other models on this task, suggesting the limit of DDPM in denoising noises that are more complex than purely Gaussian noises.

4.5 Generalization to Out-of-Distribution Data

As the DDRM work suggests, DDRM could generalize well to out-of-distribution images when it is trained on a multi-modal dataset (e.g., ImageNet [15]). Our intuitive explanation of this generalization is that DDPM could denoise any feature (e.g., the texture of an object) that is in the distribution of the training dataset. For example, the feature of furry texture learned from images of cats could also be applied to denoising a picture of a baboon. When DDPM is trained on a multi-modal dataset, it could process any image whose major components are in the distribution of the training dataset. To verify our conjecture, we test if the DDPM trained on images of churches could denoise images of bridges (retrieved from Google Maps) and owls (retrieved from ImageNet [15]) with different levels of Gaussian noises.

In terms of the quantitative metrics, DDPM could denoise the bridge images almost as good as church images for all levels of noise we tested on. However, notice that the denoising of the bridge image with the noise of $\sigma = 0.4$ (Table 7) failed to preserve the semantics of the river bank behind the bridge. The river bank area was interpreted as a component of a church and is denoised into a church-roof-like object. This example indicates that DDPM has the tendency to fit the input image into the image distribution it is trained on.

DDPM also performs well on the owl image when the noise level is low, and the high-frequency areas (e.g., the stripes on the owl's face) are preserved by DDPM. There is a nosedive when the noise level rises and the number of required denoising steps increases. Not only the details of the owl are lost, but the original structure is also poorly maintained in the denoising process when $\sigma = 0.4$. In this case, the PSNR-HVS-M and SSIM of the denoised results are both lower than the one of the church images and the bridge image.

5 Limitation and Future Work

Due to limited computing resources, we did not train a DDPM model but instead used a pretrained model that only takes in images of the size 256 × 256 pixels. Therefore, we could only compare DDPM and MPRNet on 256 × 256 images, which have much lower resolution than the images demonstrated in [8]. A comparison between DDPM and MPRNet on denoising high-resolution images could be done in the future to provide a more integrated evaluation of the denoising ability of DDPM.

The qualitative analysis of the denoised results was only made by the authors, which means that the qualitative analysis could succumb to the confirmation bias of the authors and be biased toward the DDPM models. The analysis would be more objective if other human judges could participate in the visual evaluation of the denoised images.

6 Conclusion

In this work, we showed that DDPM could perform denoise very well on in-distribution images with Gaussian noises, especially when the noise level is high. DDPM presents a stronger performance in denoising than bilateral filters and MPRNet for noticeable noises, especially when the image contains high-frequency information. However, if the noise level is high enough to interfere with the semantics of the image, then DDPM might just ignore the original information and denoise the image toward the class of images it is trained on.

TABLE 7: Evaluation of DDPM on Out-of-Distribution Images.

Original		$\sigma = 0.05$		$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.4$	
		PHM	SSIM	PHM	SSIM	PHM	SSIM	PHM	SSIM
	Bridge Noisy	32.70	0.86	26.30	0.62	20.25	0.35	13.88	0.17
	Bridge Denoised	34.13	0.91	28.07	0.87	22.66	0.78	16.86	0.70
	Owl Noisy	33.36	0.89	26.56	0.66	20.13	0.38	14.21	0.18
	Owl Denoised	33.00	0.92	26.80	0.87	20.78	0.72	14.70	0.59

In terms of generalization, DDPM might not denoise well when the images have a mix of both Gaussian and Poisson noises. The influence of Poisson noises is more obvious when the noise levels are low. DDPM can also denoise out-of-distribution noisy images and output clear images for low noise levels, yet it might break the original structure if the noise level is high and requires more steps to denoise.

Acknowledgements

We would like to thank the instructor David Lindell and the project mentor Shayan Shekarforoush for offering feedback and related material, which greatly helped with the proceeding of this work.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[2] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin, "On between-coefficient contrast masking of dct basis functions," in *Proceedings of the third international workshop on video processing and quality metrics*, vol. 4. Scottsdale USA, 2007.

[3] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[4] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1867–1888, mar 2020. [Online]. Available: https://doi.org/10.1007%2Fs11263-020-01303-4

[5] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8162–8171. [Online]. Available: https://proceedings.mlr.press/v139/nichol21a.html

[6] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," 2022. [Online]. Available: https://arxiv.org/abs/2201.11793

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.

[8] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *CVPR*, 2021.

[9] P. Esser, "Pytorch pretrained diffusion models," https://github.com/pesser/pytorch_diffusion, 2020.

[10] S. K. B. K., "Image denoising based on non-local means filter and its method noise thresholding," *Signal, Image and Video Processing*, vol. 7, pp. 1211–1227, 11 2013.

[11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *CoRR*, vol. abs/1608.03981, 2016. [Online]. Available: http://arxiv.org/abs/1608.03981

[12] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: construction of a large-scale image dataset using deep learning with humans in the loop," *CoRR*, vol. abs/1506.03365, 2015. [Online]. Available: http://arxiv.org/abs/1506.03365

[13] A. K. Boyat and B. K. Joshi, "Image denoising using wavelet transform and wiener filter based on log energy distribution over poisson-gaussian noise model," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, 2014, pp. 1–6.

[14] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1692–1700.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2014. [Online]. Available: https://arxiv.org/abs/1409.0575

APPENDIX

Fig. 5: Quality of the denoised images to t.

(a) PSNR-HVS-M to t for images with different levels of noise when $n_s = t$.

(b) SSIM to t for images with different levels of noise when $n_s = t$.

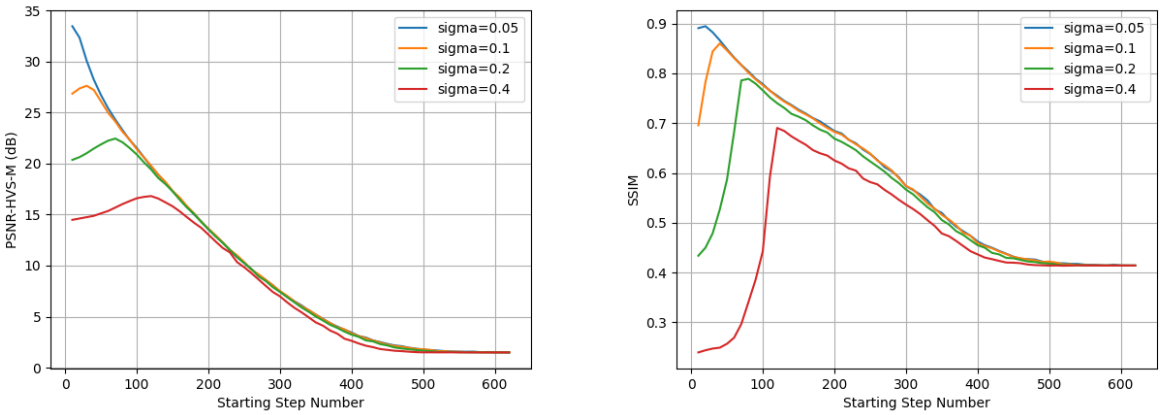


TABLE 8: Evaluation of DDPM on Gaussian-Poisson noises.

-	$\sigma = 0.05$		$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.4$	
Metrics	PHM	SSIM	PHM	SSIM	PHM	SSIM	PHM	SSIM
Gaussian-Only Noisy	33.23	0.87	26.57	0.68	20.12	0.41	14.29	0.22
Gaussian-Only Denoised	33.46	0.90	27.87	0.86	22.57	0.78	16.85	0.69
Gaussian-Poisson Noisy	32.24	0.85	25.38	0.63	18.60	0.36	12.43	0.18
Gaussian-Poisson Denoised	32.52	0.89	26.39	0.81	20.11	0.67	12.60	0.52

TABLE 9: Evaluation of DDPM denoising images with realistic noises.

Original	Noisy			BF			MPRNet			DDPM		
-	PSNR	PHM	SSIM	PSNR	PHM	SSIM	PSNR	PHM	SSIM	PSNR	PHM	SSIM
	16.87	17.71	0.44	19.06	17.44	0.68	19.46	**18.32**	**0.70**	**19.50**	17.94	0.68