Guiding Deep Image Prior with Traditional Image Recovery Methods

Jiachen Xu, Bingjie Shen, and Jingtao Zhou

Abstract—Without training processes facilitated with large image datasets, recovering a standalone noisy image can be challenging. In this project, we plan to investigate the effect of combining traditional rule-based recovery algorithms with Deep Image Prior, a recent advancement seen in neural networks image denoising, forming a new denoising pipeline. We prove that the combined method outperforms the individual results for a diverse set of images. Additionally, we study the early stopping point mechanism under the Deep Image Prior framework. We cannot observe a clear correlation between the stopping point and the intrinsic content frequency of images, and factors that may affect the image restoration speed are yet to be discussed.

Index Terms—Computational Photography, Deep Image Prior, Image Denoising, Image Deblurring.

1 Introduction

Due to the hardware limitation of most modern digital cameras, noise and blurriness are prevalent in images. The battle of dealing with these two elements is a long and arduous journey, and the emerging technologies around machine learning and deep neural algorithms and techniques are bringing it to the next level. In recent years, we have witnessed neural networks that could denoise an image from a completely noisy input or deblur from a single color image, and still produce a sensible result. This remarkable ability has extended beyond image recovery and led to generative models that could produce artwork, attracting substantial media attention and participants from outside the field [1], [2].

However, most of these solutions generally require ground-truth images in training, where the trained models approximate the original image using patterns observed from other images in the entire image set. This pipeline is ideal for general-purpose images but sets barriers to recovering images in a specific field where no ground-truth images are available. Traditional denoising and deblurring methods have long been dominant in this specific task, but we have also seen substantial progress in the field of deep learning. In this project, we intend to leverage both the newest technologies in neural networks and traditional rule-based methods to form a pipeline that can recover images without training data. The main contributions of this paper are summarized as follows:

1) We proposed a novel mechanism for combining DIP with traditional image recovery methods, utilizing denoising and deblurring on the corrupted input image to enhance the model performance.
2) We provided a performance summary of different preprocessing methods when used as guidance of the DIP framework.
3) We conducted an investigation on how different input images, recovery methods, and sigma values affect the highest PSNR and optimal stopping point of the DIP training process.

2 Related Work

Ulyanov et al. [3] proposed a powerful and effective method for imaging processing called Deep Image Prior (DIP). Our project will be mainly based on evaluating and extending the DIP algorithm by using different image preprocessing methods. DIP makes use of a deep convolutional neural network to perform different tasks, such as denoising, inpainting, super-resolution, and so on. DIP itself does not require any information about the ground-true image. It takes in a randomly generated noisy image as the input, and it treats the corrupted image as our target to optimize and approach to. During the learning process, a clear image will be learned first before the corrupted image. Different from other neural network-based denoising methods, DIP itself is a non-training process that requires zero training data and validation data. The whole process treats CNN as the regularizer and thus is an optimization problem in general. Depending on the fact that noise is more reluctant for the CNN to learn since "the parametrization offers high impedance to noise and low impedance to signal." [3], we can use proper methods of early stopping to prevent the network from overfitting. The parameters will be approaching a clear image in the process of optimizing.

Several methods were investigated to stop DIP iterations before fitting to noise and thus prevent from overfitting. Many early stopping strategies require the monitoring of PSNR which needs clean images as input. However, in real-world examples, clean images can be hard to retrieve, and it slightly contradicts the objective of DIP. In 2021, Jo et al. [4] designed an early-stopping technique which requires no

- *J. Xu is with the Department of Computer Science, University of Toronto, Canada.*
  E-mail: jiachen.xu@mail.utoronto.ca
- *B. Shen is with the Department of Computer Science, University of Toronto, Canada.*
  E-mail: bingjie.shen@mail.utoronto.ca
- *J. Zhou is with the Department of Computer Science, University of Toronto, Canada.*
  E-mail: jingtao.zhou@mail.utoronto.ca

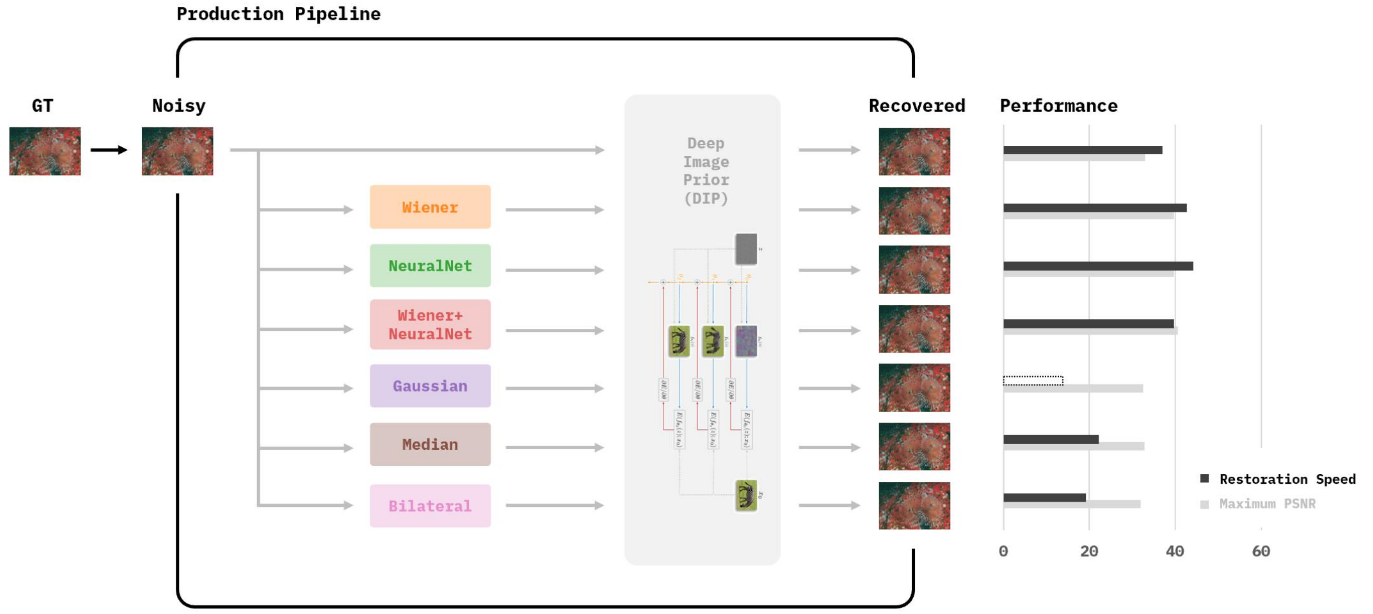

Fig. 1. The combined image restoration pipeline.

clean images. It utilizes the DIP model complexity with respect to the effective degrees of freedom which can be used as a model overfitting measurement. In general, calculating the degrees of freedom requires the ground truth image. However, the degrees of freedom can be approximated without clean images if the noise follows a Gaussian distribution. In our paper, we will mainly investigate the relationship between the early stopping point and the input image local features.

At the same time, many extensions can be made based on the idea of DIP. Bredell et al. [5] came up with an idea that uses Wiener deconvolution to guide DIP for better performance on image deblurring. The intuition behind this is that deconvolution is better at reproducing high-frequency artifacts and DIP is the opposite. Combining these two tools indeed produces better performance and stability. We may want to extend this idea and use different methods of denoising and deblurring to compare the results.

3 Proposed Method

Our work is fundamentally based on the use of Deep Image Prior (DIP) which was proposed by Ulyanov et al. [3]. As discussed in Section 2, DIP uses the original corrupted image as the target and then performs training on a randomly generated noisy image. Instead of using the original corrupted image, in our work, we attempt to perform several image preprocessing methods on the corrupted image first to remove the noise and blur and use them as the target of the DIP model. Three traditional denoising methods, Wiener Deconvolution, and a pretrained Neural Network is used as our image preprocessing techniques. We will mainly record two performance metrics which are our two objectives: the maximum PSNR of the reconstructed image with respect to our ground-true image, and the number of iterations required to reach the maximum PSNR before overfitting. This extended pipeline is shown in figure 1.

3.1 Background - Deep Image Prior

Deep Image Prior (DIP) is treated as the basis of our project which utilizes a deep convolutional neural network to perform several image reconstruction tasks without the use of any training data. This model is essentially based on the general idea of minimization:

$$x^* = \underset{x}{\operatorname{argmin}} E(x; x_0) + R(x) \tag{1}$$

where $E(x; x_0)$ is the task-dependent loss function, $x_0 \in \mathbb{R}^{3\times H\times W}$ is our target corrupted image, and $R(x)$ is the regularizer which captures the general pattern of an image to prevent from overfitting. The returned optimal solution $x^* \in \mathbb{R}^{3\times H\times W}$ is our desired clean image.

A popular choice and example of the regularizer term $R(x)$ is Total Variation (TV) which promotes a sparse gradient of the image; however, in DIP, an implicit prior captured by a neural network serves as a parametrization $x = f_\theta(z)$. Here, z is a randomly initialized 3D tensor that has the same dimension as x, and θ is thus the parameter of our neural network. This neural network parametrization can be explained as follows:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left\| f_\theta(z) - x_0 \right\|^2, x^* = f_{\theta^*}(z) \tag{2}$$

This process can be done using traditional optimization techniques such as gradient descent to obtain the optimum θ^*. Thus, the reconstructed image can be easily obtained by $x^* = f_{\theta^*}(z)$. Here, the loss function is the l2 norm capturing the l2 distance between the target image x_0 and the image returned from the network [3].

3.2 Image Recovery Methods

We used 6 image recovery methods in our work, and they serve different purposes in the image reconstruction

Fig. 2. Three Ground Truth Images Used in Task 1 and Task 2.

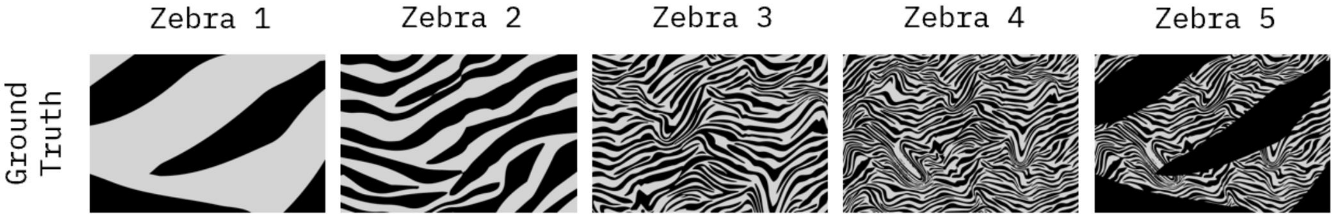

Fig. 3. Five Ground Truth Images Used in Task 2 Followup Experiments.

process. As in the real-world scenario, images can be both blurry and noisy which is hard to quantify; in our work, we prepared our corrupted image first by applying a blurry kernel on its Fourier domain and then adding a random Gaussian noise with three different noise levels: $\sigma \in \{0.01, 0.02, 0.05\}$. In order to eliminate the effect of both blurriness and noisiness, methods that are designed solely for deblurring, solely for denoising, and for both deblurring and denoising are all taken into account.

3.2.1 Traditional Denoising Methods

For the sole denoising purpose, we utilize three basic and traditional denoising methods which are Gaussian Filtering, Median Filtering, and Bilateral Filtering.

Gaussian Filtering is a naive denoising method that uses a Gaussian low-pass filter that averages pixel values in the local neighborhood. The way of averaging is based on a Gaussian distribution, by replacing the noisy pixel with surrounding pixel values, it reaches its goal of denoising for some instances. In our example, we use a Gaussian filter with a constant sigma value of 1.

Median Filtering serves a similar role as Gaussian Filtering. However, unlike Gaussian, Median Filtering is a nonlinear method that takes the median pixel value within the local neighborhood for every pixel [6]. In our work, we applied median filtering with a window of size 3.

Bilateral Filtering is based on the idea of Gaussian smoothing and is a more powerful non-linear smoothing filter than the previous two [7]. Not only considering the spatial distance between pixels but the distance of intensities is also taken into account. Therefore, only pixel values with similar intensities will be averaged, and thus it preserves edges information while smoothing.

3.2.2 Wiener Deconvolution

The method we chose to perform image deblurring is Wiener Deconvolution [8]. In our case, it uses an estimated blurring kernel as the point spread function and performs inversing techniques. Unlike basic inverse filtering which does not consider the potential noise level in the original image, Wiener deconvolution handles the noise issue by considering the signal-to-noise ratio (SNR) in its algorithm. If no noise exists, the SNR is infinitely high, and Wiener deconvolution is the same as inverse filtering in this specific scenario. However, it turns out that the performance of Wiener filtering has a tendency to decrease as the noise level gets higher and noises are largely preserved; therefore, in our work, additional denoising tasks should be performed.

3.2.3 Denoising and Deblurring Methods

As mentioned previously, in a real-world setting, both noisiness and blurriness should be handled. To address this issue, we used a basic pre-trained convolutional neural network where U-Net is selected as the network architecture. This U-Net is trained to perform both denoising and deblurring and is based on images with noisy levels $\sigma \in [0.005, 0.01]$. However, in practice, we'll also try larger noisy levels and compare their performance.

Apart from using a single neural network to perform both tasks, we also utilize a hybrid technique that combines the Wiener Deconvolution and the neural network that learns to denoise. More specifically, the deblurred images using Wiener Deconvolution are passed to the neural network to train it to learn denoising solely. Since Wiener is sensitive to noise, adding a neural network for denoising should yield a better result.

TABLE 1
Performance of selected methods using a fixed image (Image 1) and three different noise levels.

	sigma = 0.01		sigma = 0.02		sigma = 0.05	
	Initial PSNR	Maximum PSNR	Initial PSNR	Maximum PSNR	Initial PSNR	Maximum PSNR
Standalone	32.12	32.92	30.40	32.88	25.23	32.75
Wiener	28.93	39.71	23.14	38.47	15.33	34.76
NeuralNet	37.39	39.64	28.06	38.26	19.11	35.47
Wiener+NeuralNet	37.79	40.65	30.32	39.73	21.14	36.56
Gaussian	32.55	32.57	32.36	32.53	31.25	32.33
Median	32.74	32.87	32.32	32.81	30.26	32.58
Bilateral	31.91	31.97	31.64	31.92	30.01	31.73

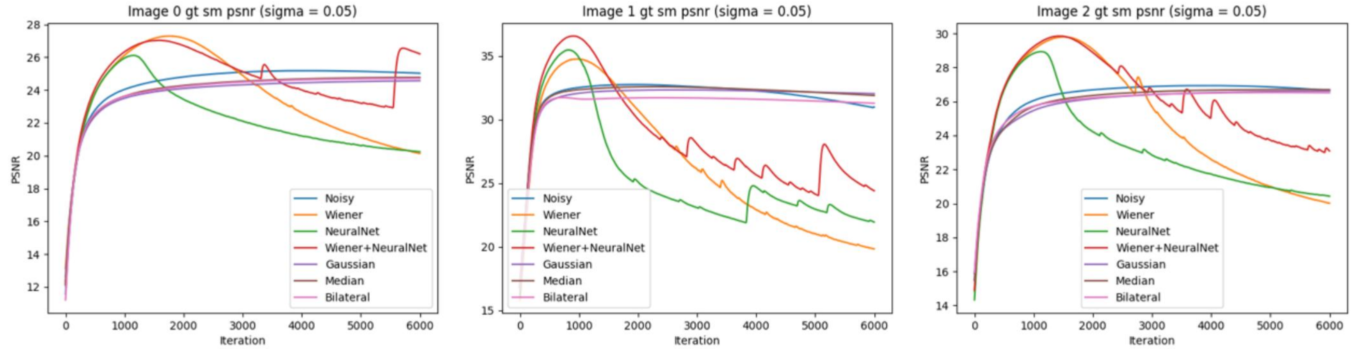

Fig. 4. PSNR Curve for selected methods under the same sigma level and different input images. Curves are grouped by input images.

4 Analysis and Evaluation

4.1 Preprocessing

Before running the DIP model with the inputs, it's necessary to apply the preprocessing methods to the corrupted image. The process can be broken down into the following steps:

1) Start with a clear image as the ground truth.
2) Blur the ground truth image and add noises to it. There are three sigma values for the noise level: [0.01, 0.02, 0.05]. This will produce three distinct noisy images.
3) For each noisy image, apply the discussed denoising and deblurring methods to it to generate six additional input images.
4) In summary, each ground truth image will lead to three versions of noisy images. Then, each noisy image will then lead to six denoised/deblurred input images. This pipeline is shown in the first section of figure 1.

4.2 Task 1: Evaluation on the maximum PSNR of the reconstructed image with respect to our ground-true image

For the first task, the goal is to investigate whether preprocessing on the noisy input will improve the training result of DIP in terms of PSNR. If so, which preprocessing method would yield the best performance?

Three images with distinct contents and levels of complexity were chosen for this task, as shown in figure 2. Those images were first processed using the preprocessing methods discussed above. Then, each input image was trained iteratively using DIP for 6,000 iterations, and the PSNR was recorded in each iteration. Table 1 shows the summarized results, while figure 4 is a visual representation of the PSNRs for the input images that were preprocessed by different methods. The image and sigma were fixed here.

4.3 Task 2: Evaluation on the number of iterations required to reach the maximum PSNR before overfitting

For the second task, the goal is to figure out the factors affecting the optimal early stop point. If multiple images were processed the same way, will the training on those images have similar early stop points? Or are there other factors affecting it?

To answer those questions, we rearranged the training results from task 1 and displayed the key numbers in table 2. The corresponding PSNR trends are shown in figure 6.

Then, we want to go one step further and generate our own input data. We want to test images that have different content frequencies but similar other properties. To achieve this, we selected an image that contains random zebra stripes and scaled them to four different zoom levels. The largest zoomed image should have the lowest frequency, and vice versa. We generated an additional image of combined lowest frequency and highest frequency output using the multiply color blend mode. Figure 3 is an overview of the 5 zebra stripe images. Next, like Task 1, those images were preprocessed and trained using DIP for 6,000 iterations, and the corresponding PSNRs were recorded. Table 3 shows the summarized results, and figure 7 demonstrates how the trends of PSNRS for different images differ during the training.

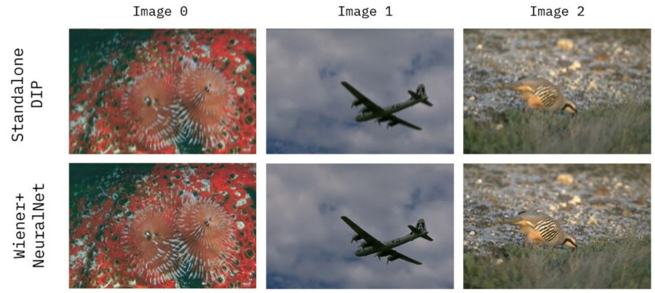

Fig. 5. Comparison of recovered images under standalone DIP and when guided by Wiener deconvolution and NeuralNet.

5 Experimental Results

5.1 Task 1: Evaluation of Image Recovery Methods and Resulting PSNRs

For Task 1, we want to quantitatively compare the performance of using different image reconstruction methods and also analyze the impact with respect to different noise levels. The resulting PSNRs are shown in table 1, and some observations of interest can be found:

5.1.1 Observation 1

Overall, we observed that inputs produced by Wiener, NeuralNet, and Wiener + NeuralNet methods tend to perform significantly better than the standalone corrupted image. This conclusion is evident in figure 5. On the other hand, inputs produced by Gaussian, Median, and Bilateral methods often have similar or even worse performance than the standalone corrupted image.

However, we also observed that sometimes the initial PSNR of the noisy input is higher than the inputs processed by Wiener, NeuralNet, or Wiener + NeuralNet methods, and it gets worse as the sigma increases.

For noise level $\sigma = 0.01$, Wiener deblurring results in an even lower initial PSNR than the original corrupted image. Also, for the other two noise levels, three deblurring techniques (Wiener, NeuralNet, and Wiener + NeuralNet) all end up with a lower initial PSNR and it gets worse as σ increases. However, for all three deblurring methods, the final maximum PSNR is still higher than the one using the unprocessed corrupted image. This seemed strange at first because we expected the preprocessing methods to improve the quality of the noisy inputs. However, after a thorough analysis of the methods and the resulting images, we revealed the underlying reason: all three methods described above were purely or partially deblurring methods. While they do a good job alleviating the impact of the blurs, the noises have been somehow magnified. This is especially obvious for Wiener, which doesn't have the additional denoising functionality that the other two have. Therefore, it's possible for the preprocessed inputs to have a lower PSNR than the noisy ones. As a comparison, there are three other preprocessing methods (Gaussian, Median, and Bilateral) that perform only denoising to the noisy input, and they tend to have high initial PSNRs.

However, after training 6000 iterations, all the deblurred inputs (NeuralNet and Wiener + NeuralNet are counted as deblurring methods although they also did denoising) outperformed the noisy input significantly, while all the denoised inputs yielded similar or even worse results. This outcome is expected, as DIP is known for being prone to image blurs, not noises. Noises won't be learned by the DIP model until later iterations, which minimizes its impact on the results. However, if the input image is very blurry, the DIP model is not capable of reconstructing the missing features, thus resulting in less ideal results. Therefore, it's reasonable to conclude that images with a high noise level but a low blurring level are more suitable inputs for DIP than images with a low noise level but a high blurring level.

TABLE 2
Restoration speed of selected guiding methods for different images at a fixed sigma level. Data was collected on the BSDS Images set.

	Optimal Stopping Point (# of iterations)		
	Image 0	Image 1	Image 2
Wiener	1743	**964**	1527
NeuralNet	1143	**823**	907
Wiener+NeuralNet	1527	**1132**	1455

TABLE 3
Restoration speed of selected guiding methods for different images at a fixed sigma level. Data was collected on the Zebra images set.

	Optimal Stopping Point (# of iterations)				
	Zebra 1	Zebra 2	Zebra 3	Zebra 4	Zebra 5
Wiener	1931	2063	1621	**1617**	1905
NeuralNet	1087	1044	**959**	1060	1415
Wiener+ NeuralNet	1258	1280	**1166**	1208	1753

5.1.2 Observation 2

As the noise level σ increases, the performance gets worse for all 7 methods (including the standalone DIP). This makes sense intuitively since with a higher noise level, it will be harder for the deep neural network to learn the detailed pattern in the images. However, for all sigma values, the trend discussed in observation 1 is preserved.

Another intriguing phenomenon is that as sigma values go higher, we can see more turbulence or fluctuations in the PSNR curve during its converging process after the peak PSNR is reached. These fluctuations shown in Figure 4 are usually manifested as a short-term rapid increase in PSNR values, followed by a gradual decline to the pre-fluctuation level. They can occur one or several times, and fluctuations of different intensities may superimpose, resulting in compound fluctuations within a period. It is worth noting that such fluctuations only appear under the method guided by Wiener and NeuralNet. At the time being, we cannot provide a reliable explanation for this turbulence, and future work may be needed to investigate the cause of such unusual patterns.

5.2 Task 2: On Factors that Affect Optimal Early Stopping Points

For task 2, results for the first set of images (Image 0, 1, and 2) show that with the same amount of noise level added, the early stopping points differ. The restoration speed for Image

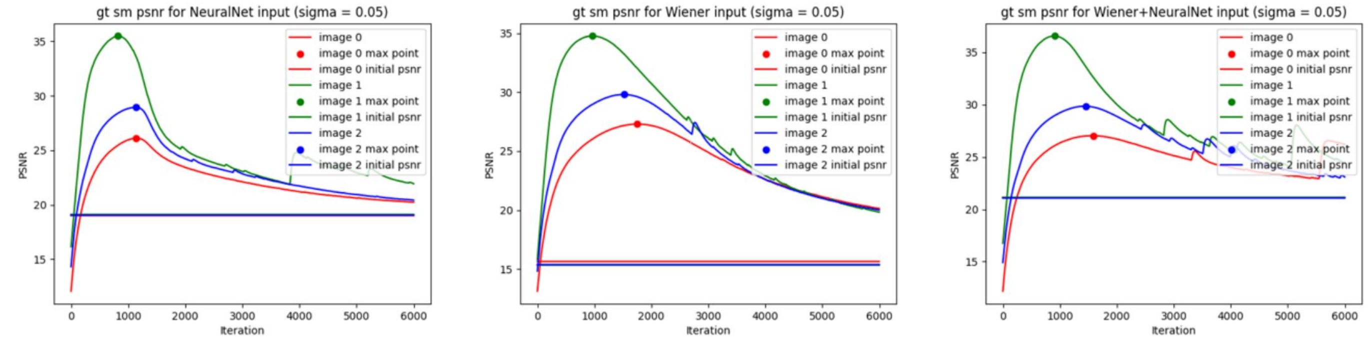

Fig. 6. PSNR Curve for selected methods under the same sigma level and different input images in BSDS image set. Curves are grouped by preprocessing methods.

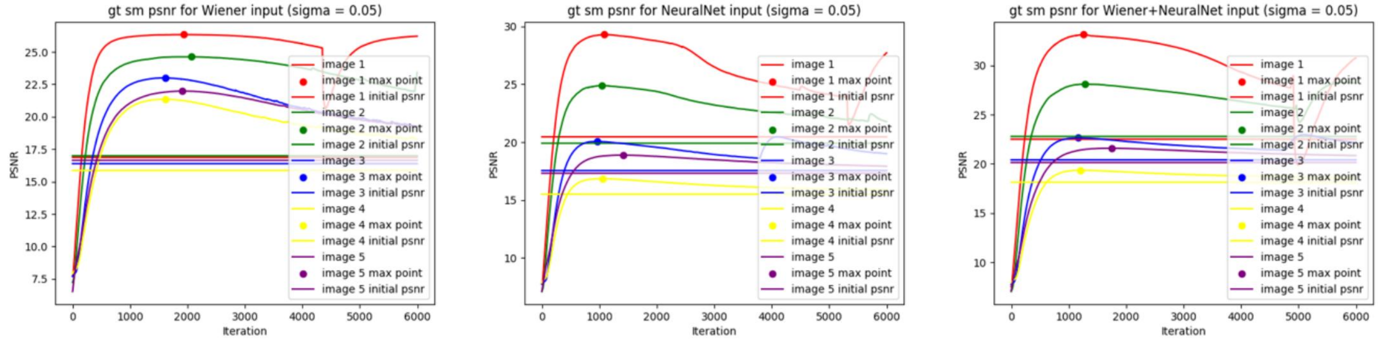

Fig. 7. PSNR Curve for selected methods under the same sigma level and different input images in Zebra image set. Curves are grouped by preprocessing methods.

0 and 2 is significantly slower than that for Image 1. From a purely qualitative perspective, the content of image 1 is less detailed and has a low overall frequency. At this point, therefore, we infer from the result that the intrinsic content frequency of an image does affect the early stopping point. This conjecture is visualized in Figure 8.

To validate this, the second set of images (Image 1, 2, 3, 4, and 5) was generated, where the images only differ in content frequency with all other possible variables controlled to a similar level. We expect to see that for images with fewer details, i.e, lower content frequency, the PSNR curve reaches peak value sooner than those with higher content frequency. However, as seen in Figure 7, the expected pattern failed to emerge.

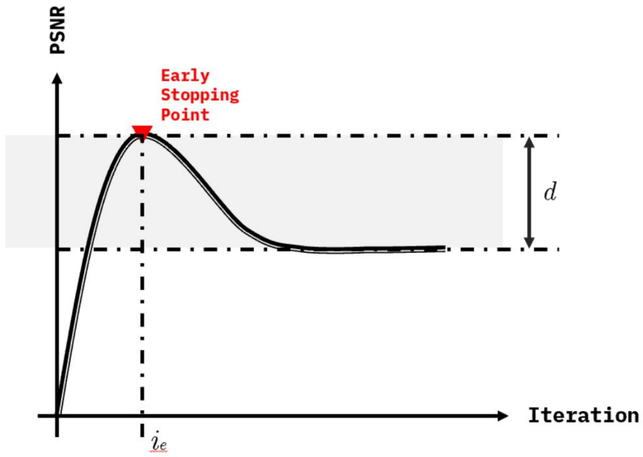

Fig. 8. The standard Deep Image Prior curve. An ideal curve should display a similar trend where it peaks in the middle, i.e, the “early stopping point”, and converge to a lower value. In our hypothesis, the difference between the peak and converged value (denoted by d) is correlated to the noise level. The time that it reaches the early stopping iteration i_e should be determined by the image content/frequency when using a fixed method.

6 Discussion

As illustrated in section 5, the three denoising methods we chose seem to have little effect on the performance of reconstruction. This might be caused by the corrupted images themselves since in our example the corrupted image we created contained both noise and blur. Although the images are denoised using those methods, the existence of blurriness still largely interferes with our performance. Therefore, if we want to dig into the effect of traditional denoising methods, we should apply denoising to an image that purely contains noise. The exact performance remains unknown. However, we should not expect a great improvement since those denoising methods involve applying a blur kernel to the image, we may end up losing some details which are essential to DIP reconstruction.

In table 3, there seem to have no obvious patterns among those 5 zebra images with different image complexities which is not what we expected. One possible assumption is that the complexity of our target image does make a contribution to the early stopping point since it can be shown clearly from table 2 that image 1 reaches its peak significantly faster than the other two images. However, image 1 has a much simpler complexity than the other two images. While in our zebra stripe examples, the difference in the level of complexity is not great enough to quantitatively reveal the pattern of early stopping points. More datasets with bigger differences in image complexity should be tested to reveal whether the pattern exists. Apart from that, since there are not enough training images in our example, we can not conclude any general and obvious patterns for the early stopping points. Therefore, larger experiments should be performed to declare a more general pattern and more accurate factors.

7 Conclusion

In recent years, the field of image denoising has seen significant progress with the advent of deep learning techniques. However, without access to large image datasets for training, the task of recovering a standalone noisy image can be a challenging endeavor. In this project, we took a deeper dive into the DIP framework. We first identified the most prominent characteristic of DIP, which is the fact that it does not require any training data to perform image recovery tasks. Then, we used seven image preprocessing methods to alter the input and then feed it into DIP. Most of these methods are traditional rule-based image manipulation algorithms that do not need training data as well. We observe that when combined with Wiener Deconvolution, denoising NeuralNet, and the combination of these two methods, both the output image quality and the restoration speed of DIP increase significantly. For the other three methods, namely Bilateral denoising, Median denoising, and Gaussian denoising, preprocessing images with them does not have an obvious impact on the performance of DIP. We discussed possible underlying explanations for this observation.

In another task, we set out to study an important concept in the DIP framework: the early stopping point. We designed a simple controlled study that plots the PSNR curve change when altering the image content frequency. From the result, we observe no clear pattern of stopping point when altering the frequency. We suggest that further investigation might be required to figure out the factors affecting the restoration speed.

Through this research, we aim to shed light on the potential of combining traditional image recovery methods with modern deep learning techniques to improve the performance of image denoising algorithms. Our findings have implications for the design and implementation of future denoising pipelines, and highlight the importance of considering the intrinsic properties of the image content in the denoising process.

In conclusion, our study demonstrates the effectiveness of combining traditional rule-based image recovery algorithms with Deep Image Prior for improved denoising performance. By leveraging the complementary strengths of both approaches, we are able to achieve superior results for a wide range of images. Additionally, our analysis of the early stopping point mechanism provides valuable insights into the underlying mechanisms of Deep Image Prior and its relevance to the denoising process.

Acknowledgments

We are deeply grateful to Prof. David Lindell for his assistance in framing the idea of this course project. Prof. Lindell insights and guidance were instrumental in helping us develop a clear and coherent approach. We would also like to express our appreciation to the dedicated teaching assistants for their support and guidance during problem sessions. Their expertise and insight helped us gain a much deeper understanding of the aforementioned image denoising and deblurring methods. We acknowledge their support and guidance throughout the semester.

References

[1] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[2] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” *arXiv preprint arXiv:2205.11487*, 2022.

[3] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.

[4] Y. Jo, S. Y. Chun, and J. Choi, “Rethinking deep image prior for denoising,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5087–5096.

[5] G. Bredell, E. Erdil, B. Weber, and E. Konukoglu, “Wiener guided dip for unsupervised blind image deconvolution,” *arXiv preprint arXiv:2112.10271*, 2021.

[6] T. S. Huang, “Two-dimensional digital signal processing ii. transforms and median filters.” in *Two-dimensional digital signal processing II. Transforms and median filters.* Springer, 1981.

[7] V. Aurich and J. Weule, “Non-linear gaussian filters performing edge preserving diffusion,” in *Mustererkennung 1995*. Springer, 1995, pp. 538–545.

[8] N. Wiener, N. Wiener, C. Mathematician, N. Wiener, N. Wiener, and C. Mathématicien, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, MA, 1949, vol. 113, no. 21.