Deep Image Prior for Single-Shot, Blind Denoising and Deconvolution

Akhilesh Balasingam and Rohan Sanda

Abstract—Denoising and deconvolution are fundamental inverse problems in image processing and computer vision, addressing issues arising from noise and blur in images. In this paper, we investigate untrained methods for solving these inverse problems, particularly blind denoising and deblurring, where prior data is unavailable. Our study explores classical methods, supervised methods, and unsupervised methods. For unsupervised methods, we implement and evaluate the Deep Image Prior (DIP) against several classical and trained baselines. Additionally, we propose adaptations to the DIP framework for blind denoising and deconvolution, utilizing TV regularization and secondary neural networks as additional priors for image reconstruction. We experiment with different architectures and regularization techniques, evaluating their performance using standard metrics such as PSNR, SSIM, and DSSIM. Results show promising outcomes for DIP-based approaches, especially when combined with additional regularization such as total variation (TV). Despite challenges related to overfitting and computational resources, our study demonstrates the potential of non-supervised methods in addressing critical, blind image processing tasks, with implications for fields like medical imaging and beyond.

Index Terms—Computational Imaging, Blind Denoising, Blind Deconvolution, Deep Image Prior

1 Introduction

Deconvolution and denoising represent critical inverse problems within image processing and computer vision, aimed at recovering sharp image details from noisy or blurred observations. These issues arise due to limitations in imaging systems, environmental factors, and transmission processes. Deconvolution seeks to remove the effects of motion or other aberrations that cause smearing or blurring in images, while denoising aims to reduce unwanted variations caused by stochasticity in sensor readings or electronic components. Deblurring and denoising techniques find wide applicability across numerous fields, including medical imaging, where clear and artifact-free images are vital for accurate diagnoses and treatment planning.

In this project, we explore non-supervised methods for solving inverse problems, specifically **blind denoising and deblurring**, where we do not have prior data. This problem is particularly relevant to fields like medical imaging – where it can be costly or harmful to the patient to retake medical readings – necessitating algorithms that can perform deconvolution and deblurring with little to no additional measurements.

We can formulate the problem using the following linear equation:

$$y = Ax + \eta$$

where $y \in \mathbb{R}^m$ are the input observations, $A \in \mathbb{R}^{m \times d}$ is the degradation operator, $x \in \mathbb{R}^d$ is the set of degraded observations, and $\eta \in \mathbb{R}^m$ is the noise, typically additive white Gaussian noise (AWGN). The degradation matrix A is problem-dependent. In this project, we consider the denoising problem, where $A = I$, and the deconvolution problem, where A represents the Toeplitz matrix corresponding to the blur kernel. In general, these inverse problems are ill-posed because $\text{rank}(A) < d$, so we often use priors to solve the following optimization problem:

$$\hat{x} = \text{argmin}_x \frac{1}{2}\|y - Ax\|_2^2 + \lambda \cdot \phi(x)$$

where the first term is our data fidelity term and the second is a regularization term that represents our prior. The data fidelity term ensures that the solution maintains similarity with the degraded measurements, while the second aspect incorporates prior knowledge and a desired property to be met, and λ is a weighting hyperparameter that balances the trade-off.

2 Related Work

There has been significant work in solving inverse problems. Prior work largely falls into one of three main categories: (1) classical, deterministic methods, (2) learned/trained methods, and (3) untrained methods.

Classical methods for denoising include algorithms such as wavelet denoising, total variation regularization, and median filtering. Classical deconvolution techniques, including Wiener filtering, Richardson-Lucy deconvolution, and blind deconvolution algorithms, aim to recover the sharpness and clarity of images by estimating the blur kernel and performing deconvolution. In contrast, BM3D, which we use as a baseline for both denoising and deconvolution in this work, exploits the redundancy present in natural images [1]. It operates in two main stages: the first stage involves collaborative filtering, where similar blocks within the image are grouped together, and the second stage applies a Wiener filter to each block to reduce noise. While these methods have demonstrated efficacy in certain scenarios, they often rely on strong assumptions about the underlying noise and

blur models, making them susceptible to inaccuracies and artifacts in real-world applications.

Unlike traditional denoising methods that rely on handcrafted features and mathematical models, DnCNN adopts a data-driven approach. It learns directly from large sets of noisy-clean image pairs, allowing it to capture complex patterns and variations in different types of images that have been corrupted with the same type and intensity of noise. The DnCNN has also shown promise in the deconvolution problem, where it must learn both the denoising and deconvolution processes. In Homework 5, we demonstrated that performing Wiener filtering on the input to the model first can improve results as the DnCNN can focus more on learning the denoising process.

While the methods discussed so far follow (1) classical methods and (2) learned methods, only the classical methods are well-suited for blind denoising and deconvolution. In order to blend the benefits of deep learning with the utility of the blind denoising problem, Ulyanov et al. introduced the concept of Deep Image Prior (DIP). The DIP uses the parametrized structure of a neural network as a prior for natural images [2]. Unlike traditional methods that rely on external datasets or handcrafted priors, DIP harnesses the architecture of a neural network itself to regularize the image reconstruction process. By optimizing the parameters of the neural network to minimize the error between the network output and a noisy observation image, DIP effectively denoises, inpaints, and reconstructs images. The versatility of DIP extends beyond denoising, offering promising applications in various image restoration tasks. Ren et al. further explored adaptations of the DIP framework for addressing deconvolution and deblurring inverse problems. Their work demonstrated that adding an additional neural network to learn the blur kernel, performance in the blind deconvolution task with DIP could be improved [3].

In this project, we adapt these methods and benchmark them against supervised algorithms (an upper bound on performance) and other deterministic algorithms discussed in class. We also explore how adding total variation (TV) regularization, simpler architectures, and varying levels and types of noise/blur affects performance.

3 Architectures

We employ and implement the following architectures in our work.

3.1 DnCNN

We use the same DnCNN architecture used in Homework 5 as a supervised baseline. Specifically, we use a network with five convolutional layers with the ReLU activation function consisting of 29507 parameters. We specifically use 64 hidden channels for each of the three hidden layers and do not use bias. The same learning rate ($\alpha = 0.001$) with the Adam optimizer and batch size of 32 is used. A stride of 3 is used in the convolutional kernels. We use the same training parameters as from Homework 5 (train for 2 epochs across the dataset).

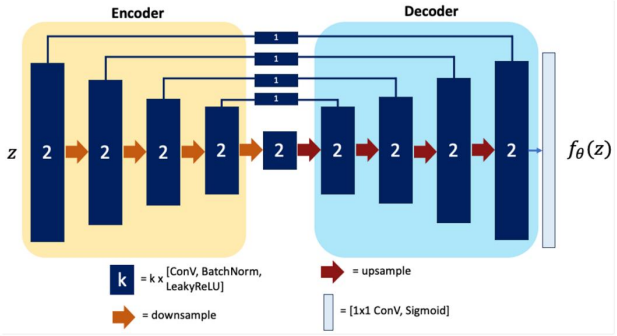

Fig. 1. U-Net Architecture

3.2 U-NET

We adapt the UNET with skip connections described in [2], as shown in Figure 1. The U-Net architecture combines the benefits of the convolutional layers present in a CNN with the encoder-decoder structure found in autoencoders.

To elaborate, the UNet architecture's symmetric encoder-decoder structure enables efficient feature extraction and reconstruction. The encoder part of the network gradually downsamples the input image, extracting abstract features at multiple scales, while the decoder part upsamples these features to reconstruct the output image. Furthermore, the use of convolutional layers in both the encoder and decoder paths ensures that the network can learn spatial dependencies and patterns present in the input image, facilitating accurate and context-aware image reconstruction.

In addition, the UNET architectures described in [2] and [3] use skip connections. These skip connections allow the network to capture both low-level and high-level features of the input image, enabling it to effectively recover fine details while maintaining global context and being more robust against the vanishing gradient problem (which results from gradients dying and hinders the network's ability to learn).

The same UNET was used in both the denoising and deconvolution task. In our implementation, we use a 5-layer encoder and decoder, consisting of 4 skip connections at every layer (see Figure 1). The LeakyReLU activation function was used, except at the final linear layer that uses a Sigmoid activation function. Our final model has approximately 2217831 parameters. After tuning hyperparamters, we use a learning rate of $\alpha = 0.001$ with the Adam optimizer. We do not employ exponential smoothing (in contrast to [2]) and an additive white gaussian noise (AWGN) regularization parameter of $r = 0.025$. Finally, we zero-pad the input to the convolutional layers. We run optimize the weights over 2000 iterations for a given image to denoise or deblur.

3.3 MLP

For our blind deconvolution reconstruction pipeline, we predict the blur kernel, using a simple 2-layer perceptron with sigmoid activation function. Similar to how we acheive better results using Weiner filtering with a DnCNN to separate the deconvolution and denoising tasks, the MLP performs a similar task by specifically trying to learn the blur kernel. The utility of adding this second network was

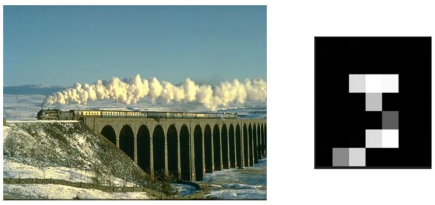

Fig. 2. (Left) Ground truth test image from BSDS300 dataset. (Right) Ground truth blur kernel from MNIST dataset

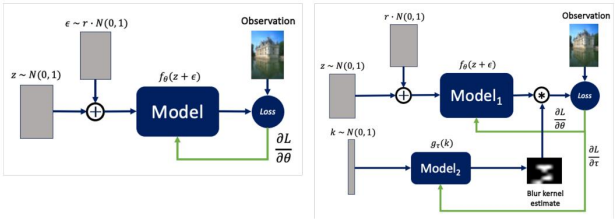

Fig. 3. (Left) Flow diagram for denoising task, (Right) flow diagram for deconvolution task

demonstrated by [3], and we follow their suggestion in our own implementation.

In our implementation, we use a learning rate of $\alpha = 0.001$, AWGN regularization parameter of $r = 0.0001$. The input to the MLP is a $x \in \mathbb{R}^{200}$ random noise vector. The output is a of dimension k x k where k is the width of the square blur kernel.

4 Methods and Theory

4.1 Dataset and Preprocessing

We use the BSDS300 dataset thoughout this work – to evaluate model performance and train the DnCNN. Specifically, we evaluate all methods on a set of five images from the "test" set that is constant across all runs. For the purposes of this report, we just report results for a single image shown in Figure 2.

With regards to preprocessing, in the case of denoising we inject noise into the BSDS300 images. In the case of deconvolution, we also blur the images using a constant blur kernel from the MNIST dataset using a method similar to that in Homework 5. The fixed blur kernel is shown in Figure 2.

4.2 Metrics

We compute the following metrics: (1) peak signal-to-noise ratio (PSNR), (2) structural similarity (SSIM), and (3) dissimilarity (DSSIM). We compute PSNR using the same methods described in class. SSIM quantifies the similarity between two images based on luminance, contrast, and structure on a scale from 0 (least similar) to 1 (most similar). SSIM provides a more comprehensive assessment of image quality compared to PSNR, capturing perceptual differences that may not be evident in pixel-wise comparisons alone. Additionally, we compute dissimilarity (DSSIM), which we calculate by first computing the image gradients via the Sobel filter and then calculating the structural similarity between the gradients of the output and the ground truth image. Together, these metrics offer a robust evaluation framework for assessing the performance of denoising and deblurring algorithms, providing insights into both fidelity and perceptual quality of the restored images.

4.3 Denoising

4.3.1 Problem Setup

In the blind denoising problem with DIP, we seek to converge to a solution for the following problem:

$$\min_\theta \text{MSE}(f_\theta(z), y)$$

where MSE denotes the mean-squared error, f_θ denotes the neural network (UNET architecture discussed above) parametrized by weights θ, and y is the noisy observation image. The model output on a Gaussian random noise "image" z is denoted by $f_\theta(z)$. We add regularization noise (corresponding to hyperparameter r) to prevent severe overfitting. This process is described in Figure 3. We run update θ over 2000 iterations.

In addition to the baseline models of BM3D, DnCNN (trained on BSDS300 dataset at same AWGN σ-level as the test image) and the new DIP model, we also experiment with adding isotropic total variation (TV) regularization [4]. TV regularization aims to enforce sparse gradients – a characteristic of natural images. Thus the new minimization problem becomes:

$$\begin{aligned} &\min_{\theta,t} \text{MSE}(f_\theta(z), y) + \lambda \|t\|_{TV} \\ &\text{subject to } \; Df_\theta(z) = t \end{aligned}$$

where $\|\cdot\|_{TV} = \Sigma_i^N \sqrt{(D_x x)_i^2 + (D_y x)_i^2}$ where D_x is the first difference matrix along the x-axis, and λ is a hyperparameter. We solve this new DIP+TV method using the alternating direction method of multipliers method (ADMM) – following the formulation used in [5] which involves updating θ in addition to an additional primal variable t. We make use of [5]'s publicly available repository and associated functions for computing the update to slack variable t. The update for θ is approximated using 50 iterations of gradient descent with learning rate $\alpha = 0.001$ with the Adam solver. After performing grid search hyperparameter tuning across many values for λ, we observe no significant improvements in MSE loss across $\lambda \in \{0.01, 0.005, 0.001, 0.0005, 0.0001\}$. We therefore choose $\lambda = 0.001$. We also employ early-stopping in calculating our results for reasons discussed in later sections. Because computing the θ update at each iteration is very computationally expensive, only 150 iterations of the DIP+TV model are run.

4.3.2 Experiments

We consider 2 types of noise: (1) additive white Gaussian noise (AWGN), as discussed in section 1, and (2) speckle noise (elementwise, multiplicative noise).

4.4 Deconvolution

4.4.1 Problem Setup

For testing blind deconvolution using the vanila DIP process, we use the same framework as for denoising. For the DIP+MLP formulation, inspired by [3], we reframe the objective as follows:

$$\min_{\theta,\tau} \text{MSE}(f_\theta(z) * g_\tau(v), y)$$

where * denotes a convolution, MSE denotes the mean-squared error, f_θ denotes the neural network (UNET architecture discussed above) parametrized by weights θ, g_τ denotes the neural network (MLP architecture discussed above) parametrized by weights τ, and y is the noisy observation image. The model output on a Gaussian random noise "image" z is denoted by $f_\theta(z)$. The model output on a Gaussian random noise "image" v is denoted by $g_\tau(v)$. Both θ and τ are jointly optimized from the same loss. We add regularization noise to prevent severe overfitting in both models. This process is described in Figure 3. We do not experiment with TV regularization in this formulation. We also employ early-stopping and learning rate-scheduling (decay factor of $\gamma = 0.1$) for reasons discussed in later sections. Both models are optimized over 2000 iterations.

4.4.2 Experiments

We experiment with two kernel sizes: $k = 7$ and $k = 13$. We fix the AWGN at $\sigma = 0.01$, so that we can observe the effects of different methods on the deconvolution problem, instead of introducing the additional factor of noise.

5 Experimental Results, Analysis, and Discussion

5.1 Denoising Results

For clarity, we show the PSNR results for 1 image, selected from the BSDS300 Dataset, in Figure 4. See above for the ground truth version.

Qualitatively, for low Gaussian noise ($\sigma = 0.1$), BM3D and DIP + TV exhibit the most realistic outcomes, closely resembling the ground truth image. BM3D effectively suppresses noise while preserving image details, resulting in clear and visually pleasing results. Similarly, DIP + TV, which integrates total variation regularization into the Deep Image Prior framework, produces sharp and well-defined images with minimal artifacts. Conversely, the output from DnCNN shows some residual noisy artifacts, particularly in the sky. Moreover, DIP without TV regularization yields relatively blurry results with prominent edge artifacts, highlighting the importance of combining regularization techniques in controlling image smoothness and preserving structural details. These qualitative observations align with our intuition, as BM3D and DIP + TV leverage sophisticated denoising strategies.

We also plot the peak-signal noise ratio and SSIM for both AWGN noise levels, for DIP+TV and DIP models, shown in Figures 5 and 6. We can see some clear trends in the PSNR plots for both noise levels. The baselines perform the best, each achieving a PSNR and SSIM of above 27 dB and 0.7, respectively. This aligns with our expectations, the DnCNN was trained on lots of data and thus has been able to capture a wide range of patterns across many images, improving its denoising capabilities. Thus we treat the DnCNN results as an "upper bound" on our expected results.

With regards to DIP, the PSNR with respect to ground truth eventually converges with enough iterations. At $\sigma = 0.1$ AWGN, the PSNR plateaus at its peak of approximately 26 dB. However, when we add more noise at $\sigma = 0.2$, we notice that the PSNR begins to falls significantly. We attribute this sharp drop to the phenomenon described in [6], wherein the model begins to overfit to the noise. We can see that the SSIM metrics follow a similar trend. We can see that, between the two noise levels, for both DIP and DIP+TV, that we capture less structural similarity between the model output and the ground truth image, at higher noise. This suggests that using early-stopping is crucial in our implementation. Thus, in both DIP and DIP + TV we employ early stopping, however we plot across more iterations to demonstrate the effect of this overfitting.

We should note that while DIP + TV achieved strong results, it was only run for 150 iterations. This is due to the fact that by using the ADMM formulation in [5] to solve the problem, we must compute 50 optimization steps in each step of the main algorithm to update θ. This is very computationally intensive. Furthermore, we did not have access to a GPU to accelerate the process. While we did experiment with reducing the number of iterations run at each step to approximate the θ update, we found that these led to worse results after 150 iterations and would need to be run for more iterations. Nonetheless, the trajectory of the PSNR and SSIM curves for the DIP + TV results suggest that using a GPU and running for more iterations would yield further improved results.

We also investigate the impact of speckle noise, a type of multiplicative noise that is prevalent in radar. Our results are summarized in Figure 8, and our qualitative results are in Figure 7. We can see that DIP and DIP+TV continue to perform well, even with speckle noise, compared to BM3D and DnCNN. This makes sense, since DIP algorithm develops understanding of the underlying structures and patterns in the images, which helps them adapt to different noise types, including speckle noise. We were surprised that the DIP and DIP+TV beat the DnCNN method (which was trained on the BSDS300 dataset with speckle noise added). We suspect that by training the DnCNN over more epochs, the more complex noise pattern of speckle noise could be learned. The low number of epochs (2) that was used for training was chosen to be consistent with the Homework 5 results, however future experimentation would involve a more robust training regime.

5.2 Deconvolution Results

We show our qualitative outputs across all deconvolution strategies for both kernel sizes in Figure 9.

Qualitatively, we can see that the Weiner + Denoise (DnCNN) performs the best on the deconvolution task, with clear and crisp edges and no artifacts. The DIP + MLP method also performs well, with comparable image quality to the DnCNN. However, the DIP method alone performs

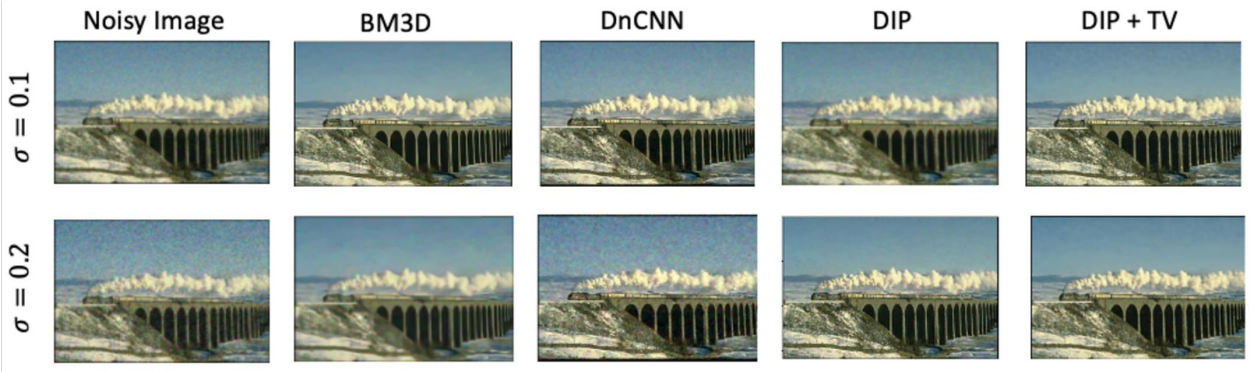

Fig. 4. Denoising Results on 1 image for AWGN

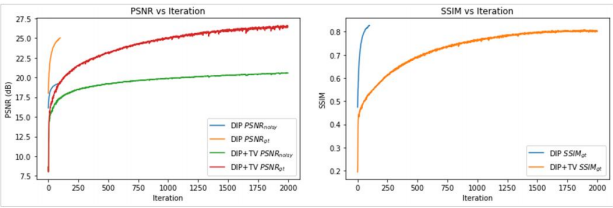

Fig. 5. PSNR vs Iteration (left), SSIM vs Iteration (right) for $\sigma = 0.1$ AWGN

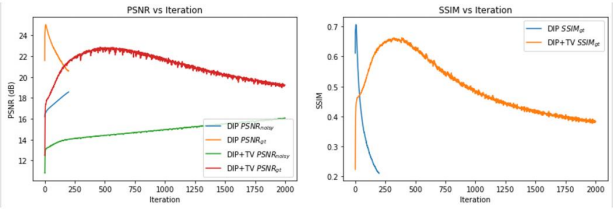

Fig. 6. PSNR vs Iteration (left), SSIM vs Iteration (right) for $\sigma = 0.2$ AWGN

poorly, with heavily blurred edges and blurring on the mountains and the clouds. This demonstrates that the DIP denoising framework on its own is not sufficient to learn both the denoising and deconvolution processes.

In comparing the DnCNN method with the Weiner + Denoise (DnCNN) approach for deconvolution tasks, it's evident that the Weiner + Denoise method outperforms DnCNN due to its combined strategy of addressing both noise and blur artifacts. Moreover, when comparing DnCNN with the DIP + MLP (interchangeably referred to as DIP + Deconvolution) method, it becomes apparent that the latter does comparably well due to the intrinsic advantages of DIP for denoising coupled with the inclusion of a separate model to learn the blur kernel. By jointly optimizing the image reconstruction and the convolutional kernel estimation, DIP + MLP effectively restores sharp details and edges that may have been lost due to blurring. So, DIP + MLP effectively addresses blurring effects while leveraging DIP's ability to infer image structures, leading to results qualitatively comparable to those achieved by DnCNN. If we had more time, we would have investigated adding TV regularization to the DIP + MLP method. We hypothesize that this would lead to even better results.

We show our loss curves for both the DIP and DIP+MLP methods for both kernel sizes in Figure 10. In both experiments, we can see that the DIP+MLP converges faster than the DIP method. We also show the PSNR results against iterations for this image for both kernel sizes: $k = 7$, $k = 13$ in Figures 11, 12. As we would expect, PSNR goes down when the blur kernel size is increased. There is a more noticeable drop in the PSNR of the DIP and DIP+MLP methods than the DnCNN and Wiener+DnCNN baselines.

We can notice a few interesting trends from these plots. First, the DIP method overfits to the noise in the noisy, blurred image. We can see this because the PSNR with respect to the noisy image continues to increase as iterations increase. In contrast, the DIP + MLP method has a peak PSNR with respect to the noisy image and then stabilizes to a PSNR of approximately 23dB with respect to the noisy image. This suggests that the DIP method alone struggles with effectively mitigating the noise present in the noisy, blurred image, leading to overfitting to the noise and limited improvements in image quality over iterations. As a result, the qualitative results of DIP are unsatisfactory due to the persistence of noise artifacts in the reconstructed images.

In the case of the DIP + MLP method, we can see that this approach reaches a peak PSNR much faster than the DIP method. A similar trend is also apparent in our SSIM results. Clearly, the addition of a second network to learn the blur kernel is effective. In contrast, the DIP method relies solely on the structure of deep neural networks to reconstruct images, which may require more iterations to achieve optimal results, especially when faced with complex image restoration tasks such as deconvolution. Without the explicit incorporation of deconvolution, the DIP method struggles to efficiently address the blurring effects present in the images, leading to slower convergence and potentially inferior results. As was stated in the Methods section, we employ early stopping in both models (though it is not needed in the MLP+DIP model which converges) to prevent overfitting to the noise.

Our results are summarized in Figure 13. We can see that the DIP + Deconvolution method (interchangeably as DIP + MLP) as well as the Weiner + Denoise methods perform the best qualitatively, however the Weiner + Denoise output

Fig. 7. Denoising Results, for Speckle Noise with $\sigma = 0.1$

Method	$\sigma = 0.1$			$\sigma = 0.2$			Speckle ($\sigma = 0.1$)		
	$PSNR_{gt}$	$PSNR_y$	$SSIM_{gt}$	$PSNR_{gt}$	$PSNR_y$	$SSIM_{gt}$	$PSNR_{gt}$	$PSNR_y$	$SSIM_{gt}$
BM3D	27.24	20.65	0.876	23.59	14.57	0.74	27.38	24.00	0.87
DnCNN	27.13	21.23	0.856	24.07	15.33	0.71	27.94	24.32	0.89
DIP	26.39	20.66	0.829	22.87	16.07	0.66	28.48	24.36	0.90
DIP+TV	25.20	19.29	0.827	23.00	16.02	0.66	29.26	24.71	0.91

Fig. 8. Results, across all methods for 1 BSDS300 image

is clearly sharper. This suggests that more robust models for approximating the blur kernel (such as deeper MLP's or additional UNETs) should be examined in the future.

5.3 Image Gradient Information

Another interesting result we investigate are the image gradients following our restoration procedures. It has been shown that natural images follow 2 tail log probability gradient distribution [7]. This distribution suggests that image gradients tend to be sparse, with a large proportion of small gradient magnitudes interspersed with occasional high-magnitude gradients corresponding to edges or boundaries between objects in the scene. Understanding and replicating this characteristic distribution in restored images is crucial for producing visually plausible and perceptually pleasing results.

Our results for denoising, with AWGN of $\sigma = 0.1$, are shown in Figure 14. Our results are also shown for deconvolution, for kernel size of 7, in Figure 15.

We can see that, although none of the reconstruction methods come very close to the ground truth log probability distribution, the model outputs are all quite close to each other near the center of the plot, indicating that both the method outputs and the original image have lots of zero-gradients – ie. they are sparse. The distribution for the models is quite similar, suggesting that our DIP outputs are in line with the baselines. This further validates the DIP and DIP+MLP results for the blind denoising and deconvolution tasks. It should be noted that these models may not closely follow the probability distribution for this specific image, but may better follow a more general probability distribution over multiple images. Future analysis would include computing the log probability distribution over multiple images and comparing that to our baselines and experimental models.

6 Conclusions, Limitations, and Future Work

In conclusion, we have explored and benchmarked the performance of the DIP and DIP + MLP models in the blind denoising and deconvolution tasks, respectively. Throughout the project, we faced numerous issues related to overfitting and compute resources. However, we ultimately achieve promising results. In the blind denoising task, we observed that DIP is effective across multiple σ-levels of AWGN noise as well as speckle noise. Furthermore, adding TV regularization to the DIP model significantly improves results – though requires more computing power to run. In the case of blind deconvolution, we observe that the DIP model alone is not sufficient to deblur an image, instead requiring a second model that can approximate the blur kernel.

In almost all cases, we find that the DIP variations are unable to meet the baselines of supervised methods. Nonetheless, the fact that the structure of a neural network can itself be leveraged as a prior is both fascinating and worthy of further exploration – especially for problems for where large training sets required for trained algorithms may not exist.

One such problem that we hope to investigate in the future is applying DIPs to the high dynamic range (HDR) image formation problem. We seek to understand whether it is possible to reduce the number of low dynamic range (LDR) images needed for HDR image formation using the DIP model. To our knowledge, very little literature has been published on this application of DIPs, and we believe that it would be an interesting way to extend these promising algorithms to new domains. It is possible that new architectures, such as generative models, may be better suited to this task. In addition, we believe it would be interesting to benchmark the performance of the DIP variations on medical imaging datasets. Complementary to this, exploring the concatenation of additional priors (beyond TV) could also be an interesting. Perhaps some hand-crafted priors are better suited for certain medical imaging datasets, and can improve the performance of vanilla DIP. In closing, the DIP algorithm presents a myriad of avenues for future exploration.

Acknowledgments

The authors would like to thank Professor Gordon Wetzstein for his advice and support.

References

[1] A. Danielyan, V. Katkovnik, and K. Egiazarian, "Bm3d frames and variational image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, 2012.

[2] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *International Journal of Computer Vision*, vol. 128, no. 7, p. 1867–1888, Mar. 2020. [Online]. Available: http://dx.doi.org/10.1007/s11263-020-01303-4

[3] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," 2020.

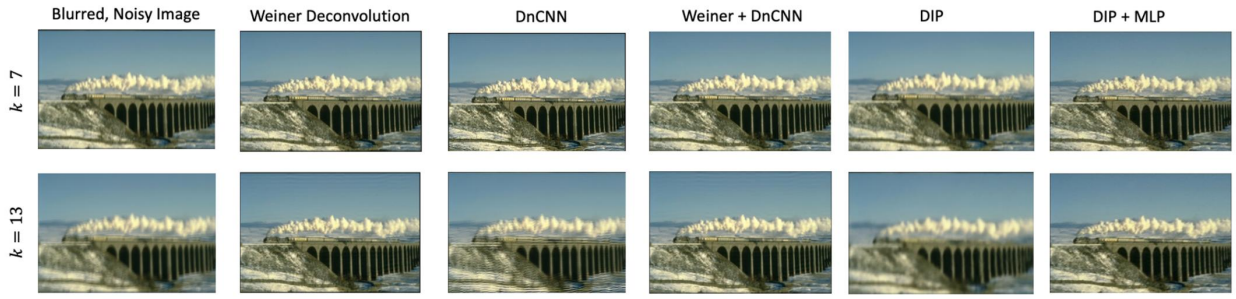

Fig. 9. Deconvolution Results, across all methods for 1 BSDS300 image

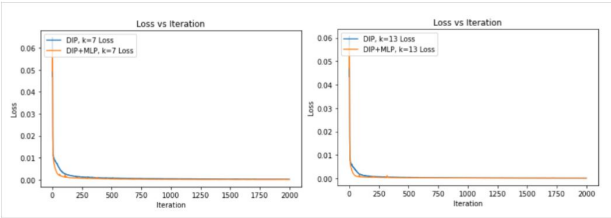

Fig. 10. Loss for Deconvolution for $k = 7$, left, and $k = 13$, right.

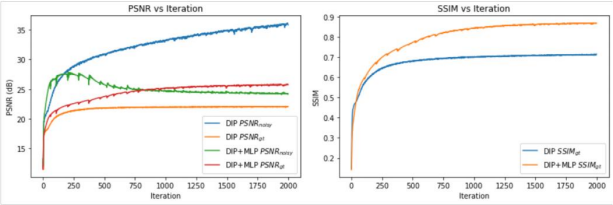

Fig. 11. PSNR vs Iteration (left) and SSIM vs Iteration (right) for $k = 7$

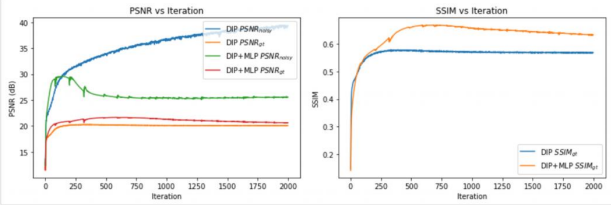

Fig. 12. PSNR vs Iteration (left) and SSIM vs Iteration (right) for $k = 13$

Method	$k = 7$			$k = 13$		
	PSNR_{gt}	PSNR_{y}	SSIM_{gt}	PSNR_{gt}	PSNR_{y}	SSIM_{gt}
Weiner	25.28	25.64	0.78	18.3	23.65	0.39
DnCNN	26.25	27.77	0.83	26.02	27.11	0.73
Weiner+DnCNN	33.39	23.51	0.94	32.06	28.42	0.92
DIP	22.09	36.16	0.71	20.32	39.49	0.58
DIP+MLP	26.21	12.15	0.87	23.57	26.63	0.77

Fig. 13. Deconvolution Results, Summary

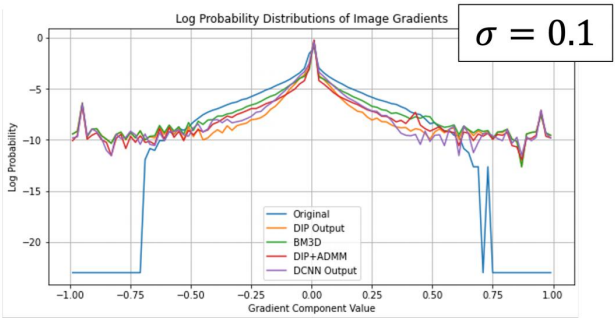

Fig. 14. Image Gradients, Denoising, $\sigma = 0.1$

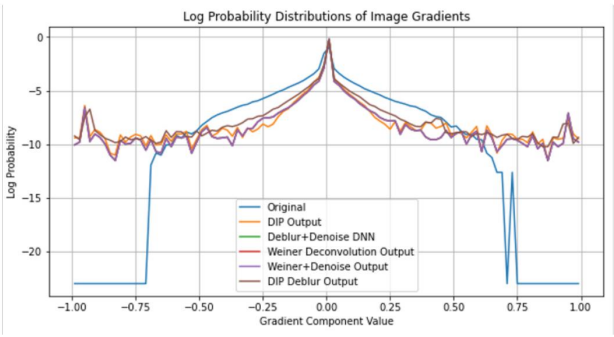

Fig. 15. Image Gradients, Deconvolution, $k = 7$

[4] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992. [Online]. Available: https://doi.org/10.1016/0167-2789(92)90242-F

[5] P. Cascarano, A. Sebastiani, M. C. Comes, G. Franchini, and F. Porta, "Combining weighted total variation and deep image prior for natural and medical image restoration via admm," in *2021 21st International Conference on Computational Science and Its Applications (ICCSA)*. IEEE, Sep. 2021. [Online]. Available: http://dx.doi.org/10.1109/ICCSA54496.2021.00016

[6] D. V. Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, "Compressed sensing with deep image prior and learned regularization," 2020.

[7] Y. Weiss and W. T. Freeman, "What makes a good model of natural images?" in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.