Deep Random Projector: Accelerated Deep Image Prior

Taihui Li[1] Hengkang Wang[1] Zhong Zhuang[2] Ju Sun[1]

[1]Computer Science and Engineering, University of Minnesota, Minneapolis, USA
[2]Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA
{lixx5027, wang9881, zhuan143, jusun}@umn.edu

Abstract

*Deep image prior (DIP) has shown great promise in tackling a variety of image restoration (IR) and general visual inverse problems, **needing no training data**. However, the resulting optimization process is often very slow, inevitably hindering DIP's practical usage for time-sensitive scenarios. In this paper, we focus on IR, and propose two crucial modifications to DIP that help achieve substantial speedup: 1) optimizing the DIP seed while freezing randomly-initialized network weights, and 2) reducing the network depth. In addition, we reintroduce explicit priors, such as sparse gradient prior—encoded by total-variation regularization, to preserve the DIP peak performance. We evaluate the proposed method on three IR tasks, including image denoising, image super-resolution, and image inpainting, against the original DIP and variants, as well as the competing metaDIP that uses meta-learning to learn good initializers **with extra data**. Our method is a clear winner in obtaining competitive restoration quality in a minimal amount of time. Our code is available at `https://github.com/sun-umn/Deep-Random-Projector`.*

1. Introduction

Deep image prior as an emerging "prior" for visual inverse problems Deep image prior (DIP) [29] and its variants [9, 21] have recently claimed numerous successes in solving image restoration (IR) (e.g., image denoising, super-resolution, and image inpainting) and general visual inverse problems [22]. The idea is to parameterize a visual object of interest, say $\boldsymbol{x}$, as the output of a structured deep neural network (DNN), i.e., $\boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z})$, where $G_{\boldsymbol{\theta}}$ is a DNN and $\boldsymbol{z}$ is a seed that is often drawn randomly and then frozen. Now given a visual inverse problem of the form $\boldsymbol{y} \approx f(\boldsymbol{x})$—$\boldsymbol{y}$ is the observation, f models the observation process, and $\approx$ allows modeling observational noise—and the typical maximum-a posterior (MAP)-inspired regularized data-fitting formulation (λ: regularization parameter):

$$\min_{\boldsymbol{x}} \underbrace{\ell(\boldsymbol{y}, f(\boldsymbol{x}))}_{\text{data-fitting loss}} + \lambda \underbrace{R(\boldsymbol{x})}_{\text{regularizer}}, \tag{1}$$

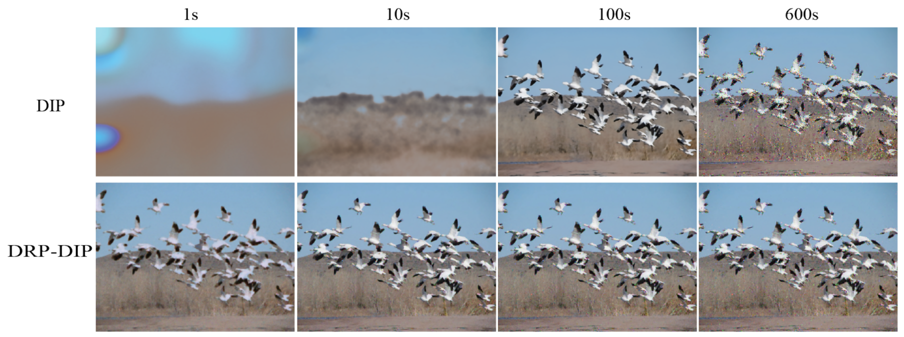
one can plug in the reparametrization $\boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z})$ to obtain

$$\min_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})) + \lambda R \circ G_{\boldsymbol{\theta}}(\boldsymbol{z}), \tag{2}$$

where $\circ$ denotes functional composition. A salient feature of DIP for IR is that they are training-free despite the presence of the DNN $G_{\boldsymbol{\theta}}$, i.e., **no extra data** other than $\boldsymbol{y}$ and f are needed for problem-solving.

DIP is not a standalone prior, unlike traditional priors such as sparse gradient [24], dark channel [8], and self-similarity [7]: *favorable results from DIP are obtained as a result of the tight integration of architecture, overparametrization of $G_{\boldsymbol{\theta}}$, first-order optimization methods, and appropriate early stopping (ES).* The $G_{\boldsymbol{\theta}}$ in practice is often a structured convolutional neural network and significantly overparameterized. Due to the heavy overparametrization, in principle, $f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})$ could perfectly fit $\boldsymbol{y}$, especially when no extra regularization $R \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})$ is present. When such overfitting occurs, the recovered $\widehat{\boldsymbol{x}} = G_{\boldsymbol{\theta}}(\boldsymbol{z})$ accounts for the potential noise in $\boldsymbol{y}$ also in addition to the desired visual content. What comes to the rescue is the hallmark "early-learning-then-overfitting" phenomenon: the learning process picks up mostly the desired visual content first before starting to fit the noise [16, 30], which is believed to be a combined **implicit regularization** effect of overparametrization, convolutional structures in $G_{\boldsymbol{\theta}}$, and first-order optimization methods [10, 12]. So, if one can locate the peak-performance point and stop the fitting process sharp there, i.e., performing appropriate ES, they could get a good estimate for $\boldsymbol{x}$.

Practical issues: overfitting and slowness Despite the remarkable empirical success of DIP in solving IRs, there



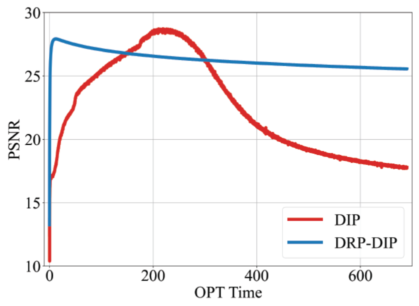


Figure 1. Comparison of DIP and our DRP-DIP for image denoising. **Left**: immediate reconstructions generated by DIP and DRP-DIP respectively at different time cutoffs. DRP-DIP can restore skeleton of the image in 1 second, whereas DIP cannot do so even after 10 seconds. **Right**: PSNR trajectories over time. Our DRP-DIP reaches the performance peak in much shorter time than DIP.

are a couple of pressing issues (see Fig. 1) impeding the practical deployment of DIP:

- **Overfitting**: Most previous works reporting the successes of DIP set the number of iterations directly, based on visual inspection and trial-and-error. Visual inspection precludes large-scale batch processing or deployment into unfamiliar (e.g., scientific imaging of unknown minuscule objects) or unobservable scenarios (e.g., multi-dimensional objects that are hard to visualize), whereas trial-and-error likely performs the tuning with reference to the unknown groundtruth object. *So, strictly speaking, previous successes mostly only show the potential of DIP, without providing an operational ES strategy [16, 22, 30].* Ideas that try to mitigate overfitting, including controlling the capacity of $G_{\boldsymbol{\theta}}$, explicit regularization, and explicit noise modeling, tend to prolong the iteration process and push the peak performance to final iterations.
- **Slowness**: For DIP to reach the performance peak from random initialization, it typically takes thousands of steps. Although the number is comparable to that of typical iterative methods for solving Eq. (1), here each step entails a forward and a backward pass through the $G_{\boldsymbol{\theta}}$ and hence is much more expensive. In fact, on a state-of-the-art GPU card such as Nvidia V100, the whole process can take up to tens of minutes for simple IR and up to hours for advanced IR tasks. The optimization slowness inevitably hinders DIP's applicability to time-sensitive problems.

These two issues are intertwined: mitigating overfitting escalates the slowness. *Thus an ideal solution is to speed up the process of climbing to the performance peak and then stop around the peak.*

Our focus and contributions Our previous works [16, 30] have proposed effective ES methods for DIP. In this paper, we tackle the slowness issue. We propose an *effective and efficient* DIP variant, called *deep random projector* (DRP), that requires substantial less computation time than DIP to obtain a comparable level of peak performance. DRP consists of three judiciously chosen modifications to DIP: (1) optimizing the DIP seed while freezing randomly-initialized network weights, (2) reducing the network depth, and (3) including additional explicit prior(s) for regularization, such as total variation (TV) regularization that encodes the sparse-gradient prior [24]. One can quickly see the superiority of our method from Fig. 1. Our main contributions include:

- proposing *deep random projector* (DRP) that integrates three essential modifications to DIP for speedup (Sec. 3);
- validating that DRP achieves peak performance comparable to that of DIP in much less time on three IR tasks (Sec. 4.2, Sec. 4.3, and Sec. 4.4), and showing that DRP is much more efficient than meta-learning-based meta-DIP for speedup (Sec. 4.6);
- demonstrating that our ES method in [30] can be directly integrated, without modification, to detect near-peak performance for DRP. Hence, we have a complete solution to both overfitting and slowness issues in DIP (Sec. 4.5).

2. Related Work

Meta-learning for speedup Meta-leaning has been used to speed up learning and computation by, e.g., learning good initializers and learning rates. Recently, this has been specialized to learning good weight initializers for SIREN [27] and DIP [34] to speed up the optimization. Compared to meta-learning that needs extra training data, our speedup method DRP needs zero extra data, inheriting DIP's advantages. Also, as we show in Sec. 4.6, meta-learning-based initializers do not have speed advantage overall despite the good initial speed.

Random projection Linear random projections have been extensively studied for randomized numerical methods, e.g., dimension reduction and sketching. For nonlinear cases, DNNs with random weights have attracted considerable interest in recent years. They can be effective random "feature" extractors for image classification [6, 13, 23, 35]. While the majority of these works focus on classification, here our random-weight networks are used for single-instance image generation.

Data-driven deep generative priors for IR This family takes pretrained generators $G_{\boldsymbol{\theta}}$, e.g., from GANs [32]. Then, either only the seed, or both the seed and the network weights $\boldsymbol{\theta}$ are optimized for IR [2, 4, 5, 21, 36]. However, training effective generators on complex real-world datasets requires enormous amounts of data, and the pretrained generators tend not to generalize well for target image outside its domain. In contrast, our DRP, similar to DIP, is a single-instance method that requires no extra training data and faces no generalizability issue.

3. Our Method: Deep Random Projector

We describe the three key ingredients of our method in Secs. 3.1 to 3.3, respectively. In Sec. 3.4, we shed light on why our method is faster than DIP by spectral bias analysis.

3.1. Optimizing the Input Seed

Alleviating inhomogeneity in learning Typically, the $G_{\boldsymbol{\theta}}$ in DIP is very deep, and gradients of the weights across different layers often have distinct magnitudes—i.e., the well-known exploding/vanishing gradient issue [20, 26] (see Fig. 2 (left)). The imbalanced gradients lead to slow optimization convergence and learning. The typical adaptive gradient methods such as ADAM ameliorate the issue, but do not entirely eliminate it. To speed up convergence, it seems natural to freeze the network weights $\boldsymbol{\theta}$ and optimize the input seed $\boldsymbol{z}$. We quickly validate this idea using image regression with DIP ($\ell(\boldsymbol{y}, G_{\boldsymbol{\theta}}(\boldsymbol{z}))$ and y here is a clean observation): Fig. 2 (right) shows that optimizing the input seed is much faster than optimizing the network weights for image regression. Therefore, *freezing network weights and optimizing the input seed is a better choice.*

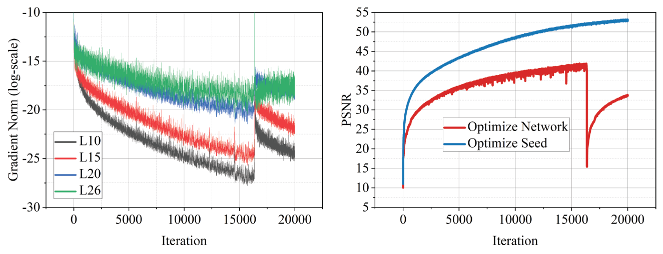


Figure 2. **Left:** gradient norm of different layers of DIP; **Right:** image regression by optimizing network or seed of DIP. When optimizing seed, we first randomly initialize the network and then freeze its weights.

Random-weight network: a free lunch For the illustration in Fig. 2, we randomly initialized network weights $\boldsymbol{\theta}$ before freezing them. While one may argue that directly taking a pretrained DNN for initialization can be another option, as we have discussed in Sec. 2, taking weights from a pretrained DNN has several foreseeable limitations. We further validate our idea with image regression: we take the generator $G_{\boldsymbol{\theta}}$ from [21] and conduct image regression but we freeze network weights and merely optimize the input seed $\boldsymbol{z}$. For the initial weights of $G_{\boldsymbol{\theta}}$, we 1) directly take weights from the pretrained model from [21] or 2) adopt the random initialization. We report the PSNR and the visualization results in Fig. 3. One can see that when we optimize the input seed $\boldsymbol{z}$ and freeze the weights of $G_{\boldsymbol{\theta}}$, randomly setting the initial weights for $G_{\boldsymbol{\theta}}$ is much better than directly taking weights from a pretrained model.

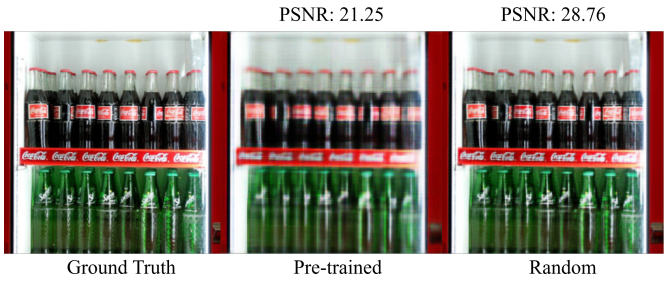


Figure 3. The result of image regression when we optimize the input seed and freeze weights of $G_{\boldsymbol{\theta}}$. (2nd) Pre-trained: we take weights from a pretrained model and set them as the initial weights of $G_{\boldsymbol{\theta}}$; (3rd) Random: we use random initialization for $G_{\boldsymbol{\theta}}$.

Based on the above discussion, we randomly initialize $G_{\boldsymbol{\theta}}$ and then freeze its weights. Typically, a DNN has batch normalization [11], and we allow its update in our experiments. Therefore, we have our initial deep random projector (DRP):

$$\min_{\boldsymbol{z}, \boldsymbol{\theta}_{BN}} \ \ell(\boldsymbol{y}, f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})), \tag{3}$$

where $\boldsymbol{\theta}_{BN}$ represents the affine parameters for batch normalization. We also make the first layer of $G_{\boldsymbol{\theta}}$ as batch normalization to make the optimization function more smooth [25].

3.2. Cutting Down the Network Depth

Using random-weight $G_{\boldsymbol{\theta}}$ and optimizing the input seed $\boldsymbol{z}$ eliminate the inhomogeneity during the learning process, which is supposed to accelerate the optimization convergence. However, after switching the role of $\boldsymbol{\theta}$ and $\boldsymbol{z}$, the per-iteration computation, which is dominated by a full forward-pass and backward-pass, is almost identical to that of the original DIP. To reduce the per-iteration computation, we propose directly cutting down the network depths. Furthermore, cutting down the depth is also expected to reduce the non-linearity of the final optimization loss and make it more smooth, which could speed up the convergence as well.

Here, we validate our thought on image denoising (see details in Sec. 4.2) with DRP Eq. (3) by using DIP as our $G_{\boldsymbol{\theta}}$ and exploring different depths of $G_{\boldsymbol{\theta}}$. We record their corresponding restoration quality and OPT time—the time to reach the PSNR peak. We report the mean PSNR and OPT time in Fig. 4. One can observe that as expected, the OPT time is drastically reduced when using a shallow backbone network—*the shallower the backbone neural network, the less the OPT time*, which is a desirable property for our model. However, the shallow backbone network has a detrimental impact on the restoration quality, which is undesirable given that we want our method to preserve the restoration quality.

Hence, the question becomes *how to bring back the restoration quality while still enjoying the benefits of shallow neural networks as our backbone networks (see Sec. 3.3).*

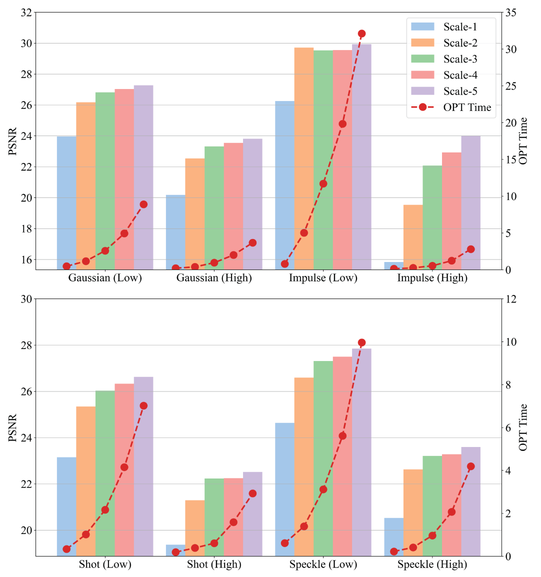


Figure 4. The PSNR and OPT time of DRP on image denoising when using a backbone network with different depths. The details of the "network scale" are provided in Appendix A.

3.3. Reintroducing Explicit Priors

We suspect that the degradation in restoration quality in Sec. 3.2 might be due to the fact that optimizing the input seed $\boldsymbol{z}$ with the frozen random-weight $G_{\boldsymbol{\theta}}$ dramatically suppresses the implicit prior induced by DIP's implicit regularization. Therefore, to bring back the restoration quality, we have to find a way to compensate. To this end, we propose employing additional explicit priors. In our paper, we employ total variation (TV) [24] due to its popularity for images:

$$\mathrm{TV}(\boldsymbol{x}) = \sum_{i,j} |\boldsymbol{x}_{i+1,j} - \boldsymbol{x}_{i,j}| + |\boldsymbol{x}_{i,j+1} - \boldsymbol{x}_{i,j}| \tag{4}$$

for any 2D image $\boldsymbol{x} \in \mathbb{R}^{h \times w}$.

Integrating Eq. (4) into Eq. (3), we finally arrive at our regularized version of DRP:

$$\min_{\boldsymbol{z}, \boldsymbol{\theta}_{BN}} \|\boldsymbol{y} - f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})\| + \lambda \mathrm{TV}(G_{\boldsymbol{\theta}}(\boldsymbol{z})), \tag{5}$$

where we use a shallow neural network $G_{\boldsymbol{\theta}}$ as our backbone to gain the benefit of fast optimization and employ the TV regularizer to retain restoration quality.

Here, we again use image denoising (see details in Sec. 4.2) for illustration and empirically compare the performance of DRP *without* and *with* TV regularization. We report the mean PSNR in Tab. 6. One can observe that: 1) adding TV to DRP significantly boosts the restoration quality, i.e., higher PSNR; 2) this observation is consistent across different noise types and different noise levels. We also compare with the performance using the TV regularization only without DIP: directly in the pixel space (see Appendix E).

Table 1. Comparison of DRP *without* and *with* total variation.

		DRP w/o TV	DRP w/ TV	Δ
Gaussian	Low	23.93	28.82	4.89
	High	20.19	24.44	4.25
Impulse	Low	26.22	31.34	5.12
	High	15.31	25.32	10.01
Shot	Low	23.19	27.86	4.67
	High	19.37	22.82	3.45
Speckle	Low	24.54	28.89	4.35
	High	20.38	24.15	3.77

3.4. Spectral Bias Analysis

Here, we explore the differences between our DRP and the DIP from the perspective of spectral bias in two settings: recovering images from noise-free images and noisy images. For each setting, we select 2 images: Baboon and Lena (Sec. 4.2). We then run each model for 300 seconds and analyze the spectral bias over the optimization trajectory. To measure spectral bias, we follow the idea in [37] and use their proposed metric—frequency band error (FBE)—that calculates the point-wise relative estimation error over the Fourier domain $|\mathcal{F}(\boldsymbol{y}) - \mathcal{F}(\widehat{\boldsymbol{x}})| / |\mathcal{F}(\boldsymbol{y})|$, and then divides the Fourier frequencies into five bands radially and computes the per-band average, where $\boldsymbol{y}$ is the given observation and $\widehat{\boldsymbol{x}}$ denotes the reconstructions by DRP or DIP. We show the evolution of the FBEs of all five frequency bands against the optimization time under noise scenario in Fig. 5 (results of noise-free scenario are shown in Appendix G). It is evident that DRP recovers all frequency bands much faster and reliably than DIP.

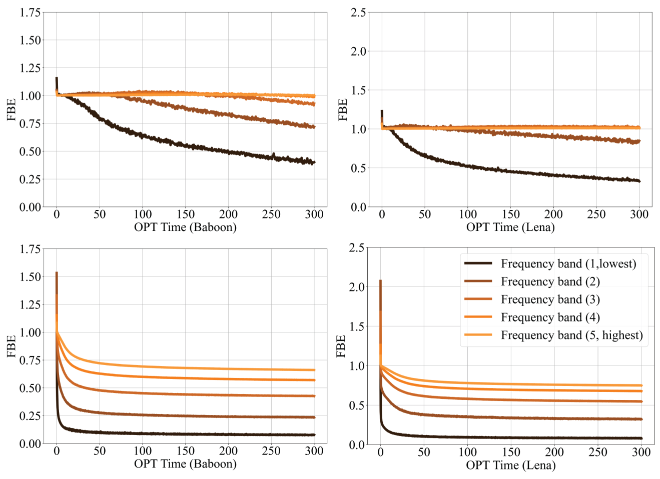


Figure 5. We compare the spectral bias of DIP (first row) and DRP (second row) under the noise scenario.

4. Experiments

4.1. Experimental Settings

In this paper, we assume that only **a single** IR instance is available each time. We consider two choices for $G_{\boldsymbol{\theta}}$: it can be a *hourglass* architecture like DIP [29] which we call it as DRP-DIP; or a *decoder* architecture like DD [9] which we call it as DRP-DD. The detailed architecture is illustrated in Appendix A. In addition, we demonstrate that state-of-the-art early-stopping methods for DIP can be incorporated directly into our methods without modification (Sec. 4.5). On the other hand, when *extra* large-scale training data is available, there are two other methods—MetaDIP [34] uses meta-learning to initialize DIP to speed up its training process and deep generative priors (DGP) [21], see also GAN-inversion methods [32], uses pretrained models on ImageNet as its generator for faithful image reconstructions. Even though it is unfair to compare our methods, which require no additional images, with MetaDIP and DGP, which heavily rely on additional training data, we still select one task from each paper and compare DRP, MetaDIP, and DGP in Sec. 4.6. In addition, we use Adam [15] as our optimizer for all experiments and introduce other detailed experimental settings in their corresponding subsection. For evaluation, we provide some qualitative vitalization; and we also provide PSNR (↑) and optimization (OPT) time (↓) as our quantitative metrics. In general, an ideal model should achieve reasonably good restoration quality within the shortest OPT time.

4.2. Image Denoising

We first conduct experiments on noise removal where only a noisy image $\boldsymbol{y}$ is given and the objective is trying to restore a clean image $\boldsymbol{x}$ from it. Therefore, we turn Eq. (5) into:

$$\min_{\boldsymbol{z},\boldsymbol{\theta}_{BN}} \|\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z})\| + \lambda \mathrm{TV}(G_{\boldsymbol{\theta}}(\boldsymbol{z})). \tag{6}$$

An example of impulse noise is shown in Fig. 6. One can see that the restored images by our methods show almost no perceptual discrepancy to others and we further show the OPT time advantage of our methods in Fig. 7, Fig. 22, Fig. 23, and Fig. 24.

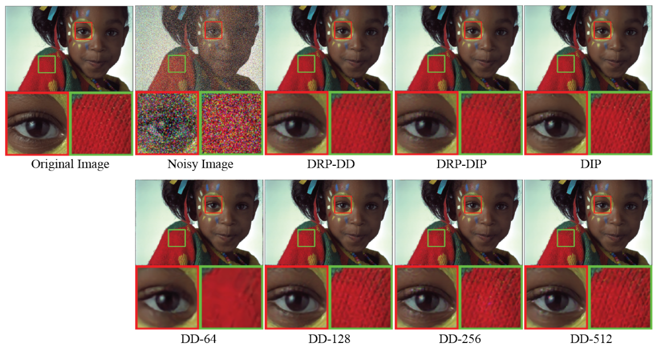


Figure 6. A visualization of image denoising on impulse noise with 30% corruption ratio. Our methods (DRP-DD and DRP-DIP) produce restorations that are visually superior to DD-64 and comparable to DD-512 and DIP.

To make our evaluation thorough, we further experiment on a standard image denoising dataset[1] with 4 different noises including Gaussian noise, impulse noise, shot noise, and speckle noise. For each noise, we test a low and high noise level, of which we follow the exact same noise settings in [16]. In addition, we use the exact same hyper-parameters for both DRP-DD and DRP-DIP for all images across all noises and noise levels: we set the learning rate as 0.1 and λ as 0.45. For DD and DIP, we use their original hyper-parameter settings while for DD, as its performance on different noises and noise levels may be sensitive to its network width, thus we in addition test different network width $\{64, 128, 256, 512\}$ which we term them as DD-64, DD-128, DD-256, and DD-512, respectively. For all models, we run them for $20K$ iterations and then report the peak PSNR and its corresponding time to reach the peak point.

Fig. 7 shows the denoising performance on shot noise (the performance for Gaussian noise, impulse noise, and speckle noise are provided in Appendix G). In terms of image restoration quality, both DRP-DD and DRP-DIP achieve competitive PSNR as that of DD and DIP; while

[1]http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_results

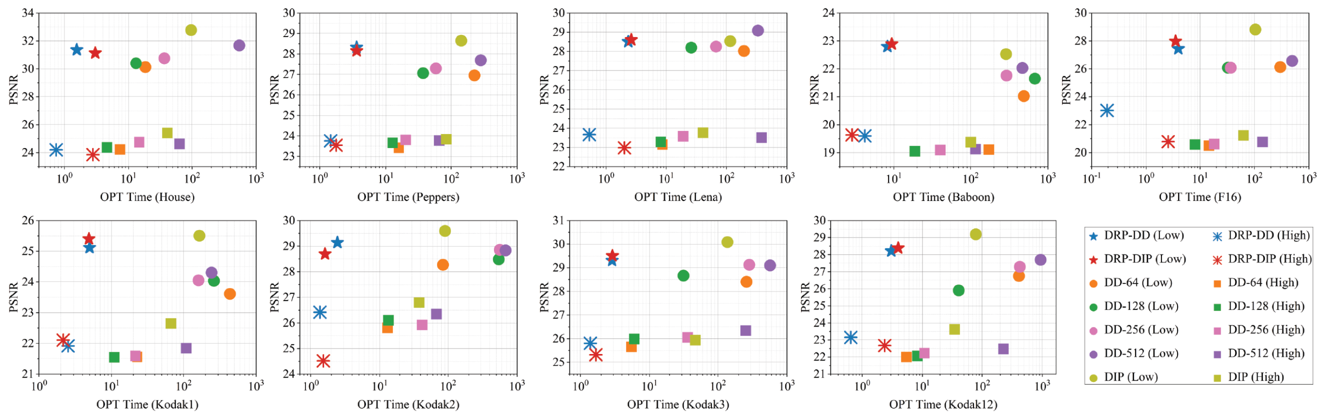


Figure 7. Quantitative comparison of image denoising on low- and high-level shot noise.

in terms of OPT time, DRP-DD and DRP-DIP show a significant advantage over DD and DIP. Moreover, the above observations are consistent across different images, noises, and noise levels.

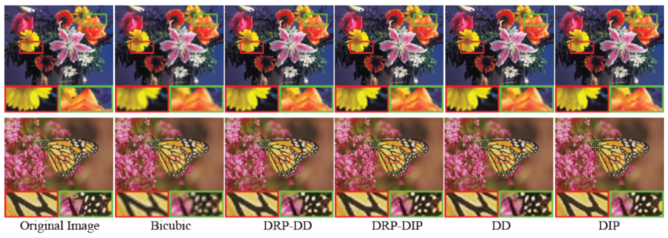


Figure 8. A visualization of image SR with $4\times$ factors. Our methods (DRP-DD and DRP-DIP) produce restorations that are visually superior to Bicubic and comparable to DD and DIP.

4.3. Image Super-Resolution

With a low resolution (LR) image $\boldsymbol{y} \in \mathbb{R}^{3\times H\times W}$ and an upsampling factor t, image super-resolution (SR) attempts to generate a corresponding high resolution (HR) image $\boldsymbol{x} \in \mathbb{R}^{3\times tH\times tW}$. In our case, DRP-DD, DRP-DIP, DD, and DIP are supposed to generate HR image $\boldsymbol{x}$. Thus, by applying a downsampling operation to the generated HR image $\boldsymbol{x}$, we hope to find its low-resolution version is exactly the given LR image $\boldsymbol{y}$. The optimization objective is the same as Eq. (5) except we downsample the restored image before calculating the ℓ_2 distance in the fidelity term:

$$\min_{\boldsymbol{z},\boldsymbol{\theta}_{BN}} \|\boldsymbol{y} - \mathcal{D}(G_{\boldsymbol{\theta}}(\boldsymbol{z}))\| + \lambda \mathrm{TV}(G_{\boldsymbol{\theta}}(\boldsymbol{z})), \tag{7}$$

where $\mathcal{D}$ is the downsampling operator, which we use the Lanczos filter in our experiments by following [29]. One should also note that we apply TV to the restored image directly without downsampling it. Similar to DIP, we consider two popular image super-resolution datasets—the Set5 [3] and the Set14 [33], and experiment with upsampling factor $t = 4$ and $t = 8$. We then compare the performance of our methods with DD and DIP. Both DRP-DD and DRP-DIP adopt the exact same hyper-parameters with the learning rate being 0.5 and λ being 0.75. For DD and DIP, we follow their original settings. We run all models for $5K$ iterations and report the peak PSNR along with the corresponding OPT time to reach the peak.

We first show the visual comparison of $4\times$ SR in Fig. 8 (the visual comparison of $8\times$ SR is provided in Appendix G), respectively. For qualitative comparison, we also provide the HR image generated by Bicubic upsampling. Visually, our methods, DD, and DIP yield better HR images when compared with the simple Bicubic upsampling while both DRP-DD and DRP-DIP are on-par with DD and DIP regarding the visual quality of generated HR images. We further report the quantitative comparison in Fig. 9 for Set5 (the result for Set14 is provided in Appendix G). As expected, our methods obtain similar PSNRs as those of DD and DIP, while both DRP-DD and DRP-DIP show a strong quantitative advantage in OPT time.

4.4. Image Inpainting

Image inpainting (IP) is another common IR task: a clean image $\boldsymbol{x}$ is contaminated by a binary mask $\boldsymbol{m} \in \{0,1\}^{H\times W}$ such that $\boldsymbol{y} = \boldsymbol{x} \odot \boldsymbol{m}$ where $\odot$ denotes the Hadamard pointwise product, and the purpose of IP is to find the missing pixels and eventually restore $\boldsymbol{x}$ given $\boldsymbol{y}$ and $\boldsymbol{m}$. Hence, we turn the optimization function Eq. (5) into:

$$\min_{\boldsymbol{z},\boldsymbol{\theta}_{BN}} \|\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z}) \odot \boldsymbol{m}\| + \lambda \mathrm{TV}(G_{\boldsymbol{\theta}}(\boldsymbol{z})). \tag{8}$$

In this task, for DRP-DD and DRP-DIP, we set the learning rate as 0.05 and λ as 0.01. Furthermore, we find that adding small perturbations ε to the network weights in each iterative process can help improve the restoration quality for our method (DIP [29] adds a small perturbation to the input

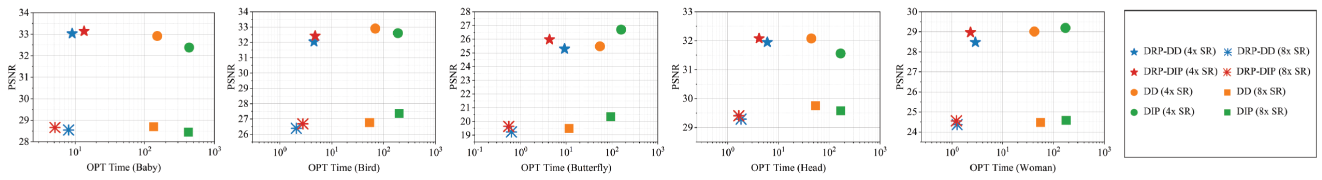


Figure 9. Quantitative comparison of image SR on Set5.

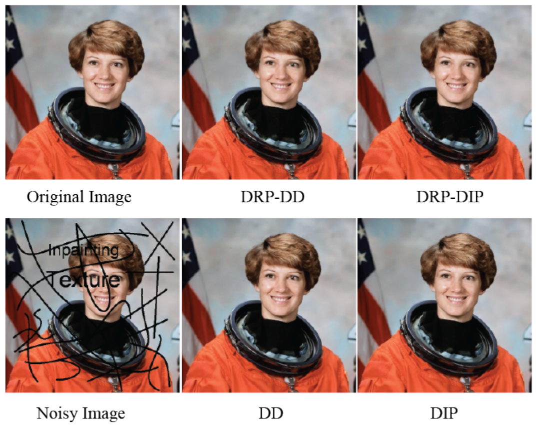


Figure 10. A visualization of IP with marks and texts. Our methods (DRP-DD and DRP-DIP) produce restorations that are visually comparable to DD and DIP.

seed $\boldsymbol{z}$). In our experiments, the ε is taken from a Gaussian distribution $\mathcal{N}(0, 10^{-3})$. For DD and DIP, we simply adopt their original settings including the network architecture and hyper-parameters.

We first experiment with our methods for text removal where a clean image is overlaid with some marks and texts, and the goal is to restore the clean image from it. Fig. 10 shows the qualitative comparison of our methods, DD, and DIP. Visually, one cannot see much difference from these restored images by different models, implying that our methods have similar power to restore high-quality images as that of DD and DIP in the task of image inpainting. We further show the OPT time advantage of our methods in Fig. 11, Fig. 26, and Fig. 27.

Now, to further investigate the advantages of our methods in terms of OPT time, we conduct image inpainting on the same inpainting dataset used in DIP [29] where the mask $\boldsymbol{m}$ is generated according to an iid Bernoulli model, with a rate of $\{10\%, 30\%, 50\%\}$, i.e., $10\%, 30\%, 50\%$ of pixels not observed in expectation, which we term them as the problem of IP-10%, IP-30%, IP-50%, respectively. Based on our empirical observation, the PSNR of these experimental models (ours, DD, and DIP) keep increasing, thus, instead of running them for a fixed number of iterations as what we have done in Sec. 4.2 and Sec. 4.3, we here run all models under a controlled OPT time (e.g., in our experiments, we cap the OPT time to be 300 seconds) and then report the change of its restoration quality in terms of its corresponding OPT time in second. The quantitative comparisons of IP-10% are shown in Fig. 11 (we provide the results of IP-30% and IP-50% in Appendix G). Although DD and DIP eventually may catch up with our methods in terms of getting similar PSNR, our methods demonstrate strong advantages over DD and DIP regarding the optimization speed, especially if one zooms into the comparisons in the first 30 second.

4.5. Early-Stopping for Deep Random Projector

We demonstrate that our methods can achieve peak performance significantly more quickly than DIP in Sec. 4.2, Sec. 4.3, and Sec. 4.4. Simultaneously, we also observe that our methods exhibit similar overfitting issues as DIP. Here, we investigate whether the existing early-stopping methods for DIP could be incorporated into our methods for combating overfitting. Among recent early-stopping methods [16, 30], we use ES-WMV [30], a lightweight and effective early-stopping method for DIP, for experimental purpose. We focus on image denoising and adopt the exact same settings in Sec. 4.2. For ES-WMV [30], we set its patience number as 300 and its window size as 50. We also follow the evaluation pipeline of ES-WMV [30] and report the PSNR gap in Fig. 12. One can observe that in most of the cases, the detection PSNR gap is less than 1 dB, implying that ES-WMV [30] is quite effective for our methods without any modifications.

4.6. Comparison with MetaDIP and DGP

Here we compare our methods, which operate in the complete absence of training data, with MetaDIP [34] and (DGP) [21], of which the performance heavily relies on exterior training images. In particular, for MetaDIP [34], we set the inner loop iterations to 20 and 50, respectively, and name them MetaDIP-20 and MetaDIP-50 accordingly. We also follow the training settings in [34] that we use 128×128 center-cropped images from the CelebA [18] for training. The meta-initializations are trained for $150,000$ outer loop iterations with the center-cropped CelebA and additive Gaussian noise at $\sigma = 25$. After that, we test DIP, MetaDIP20, MetaDIP50, and our method (DRP-DIP) on CelebA [18] and CBSD68 [19] with low- and high-level

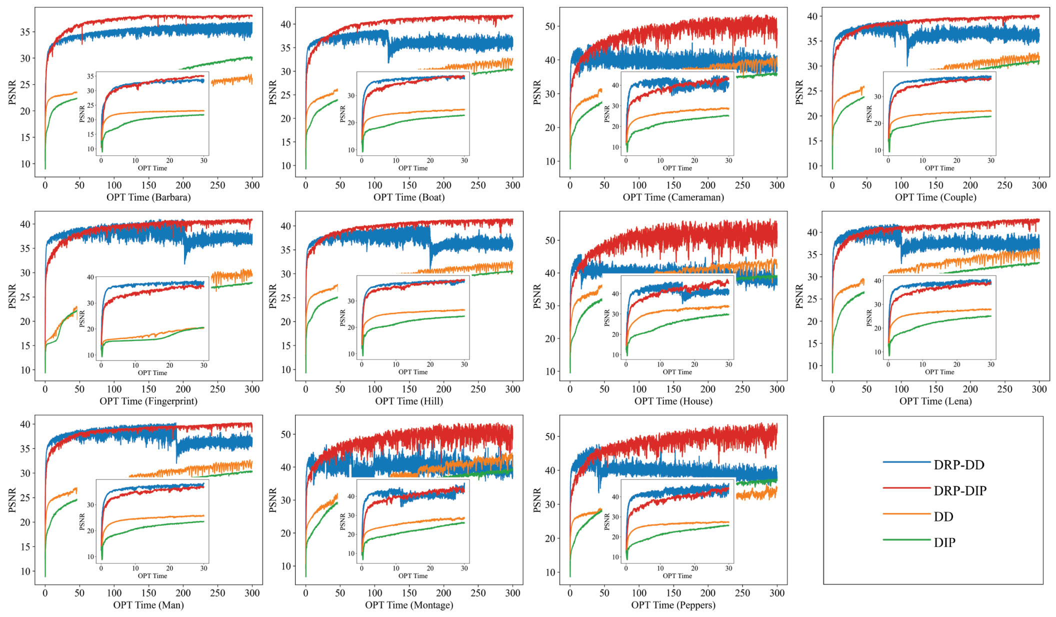


Figure 11. Quantitative comparison of IP-10%. The small sub-figure shows the comparison in the first 30 second.

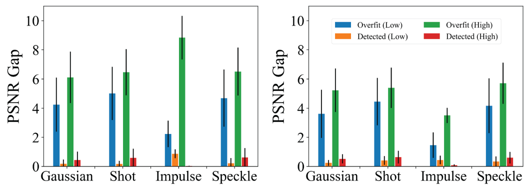


Figure 12. The performance of applying the existing early-stopping method to our methods—DRP-DD (left) and DRP-DIP(right)—in various noise types and noise levels. "Detected" and "Overfit" represent the performance with and without the early-stopping method, respectively.

Gaussian noise and report the mean peak PSNR and the corresponding OPT time in Tab. 2. One can observe that although DIP with meta-learning has access to large-scale datasets, it does not help much in either boosting restoration quality or reducing optimization time; and our DRP-DIP achieves the best performance in this setting.

For DGP [21], we compare our method with it on image inpainting. To investigate and compare the generality of both methods, we test images outside of ImageNet, on which the DGP pretrained GAN has been trained, and use the 128×128 center-cropped images from the CelebA-HQ [14] with the generated random binary mask as Sec. 4.4. Following the similar evaluation pipeline in Sec. 4.4, we adopt 3 different time-checkpoints (1/10/100 seconds) and reported the averaged PSNR in Tab. 3. One can observe that: 1) the DGP cannot generate restorations with high PSNR, probably it suffers from the data distribution shift; 2) our DRP-DIP obtains a reasonable PSNR very quickly (in just 1 second) and consistently outperforms DIP and DGP in other time checkpoints, suggesting the superiority of our methods.

Table 2. The comparison of DIP, metaDIP, and DRP-DIP for image denoising. We report results in the form: mean PSNR (OPT time). Higher mean PSNRs are in red and less OPT time is in blue.

	CelebA		CBSD68	
	L	H	L	H
DIP	27.27 (9.44)	23.04 (3.95)	26.05 (9.44)	22.55 (3.67)
MetaDIP20	27.27 (7.59)	23.02 (3.24)	26.16 (8.89)	22.54 (3.99)
MetaDIP50	27.31 (7.01)	23.09 (2.69)	26.10 (8.50)	22.51 (3.39)
DRP-DIP	27.99 (0.49)	23.70 (0.18)	26.86 (0.73)	23.07 (0.34)

Table 3. The comparison of DIP, DGP, and DRP-DIP for image inpainting. The averaged PSNR is reported.

	IP-10%			IP-30%			IP-50%		
	1s	10s	100s	1s	10s	100s	1s	10s	100s
DIP	18.02	26.50	35.00	18.01	26.50	34.00	18.05	26.45	34.00
DGP	9.94	16.00	25.00	8.77	13.00	22.00	7.26	9.00	14.00
DRP-DIP	**32.98**	**38.72**	**42.00**	**31.80**	**36.26**	**38.00**	**29.95**	**33.59**	**35.00**

Acknowledgements HW, ZZ, and JS are partially supported by NSF CMMI 2038403. The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper.

References

[1] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 15

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William S. Peebles, Bolei Zhou, Hendrik Strobelt, and Antonio Torralba. Seeing what a GAN cannot generate. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4501–4510. IEEE, 2019. 3

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In Richard Bowden, John P. Collomosse, and Krystian Mikolajczyk, editors, *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–10. BMVA Press, 2012. 6

[4] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE Trans. Neural Networks Learn. Syst.*, 30(7):1967–1974, 2019. 3

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3

[6] Ángel López García-Arias, Masanori Hashimoto, Masato Motomura, and Jaehoon Yu. Hidden-fold networks: Random recurrent residuals using sparse supermasks. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, page 205. BMVA Press, 2021. 3

[7] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 349–356. IEEE Computer Society, 2009. 1

[8] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2341–2353, 2011. 1

[9] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 1, 5, 11

[10] Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 1

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. 3, 15

[12] Gauri Jagatap and Chinmay Hegde. Algorithmic guarantees for inverse imaging with untrained network priors. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14803–14813, 2019. 1

[13] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 2146–2153. IEEE Computer Society, 2009. 3

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 8

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[16] Taihui Li, Zhong Zhuang, Hengyue Liang, Le Peng, Hengkang Wang, and Ju Sun. Self-validation: Early stopping for single-instance deep generative priors. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, page 108. BMVA Press, 2021. 1, 2, 5, 7

[17] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S. Kamilov. Image restoration using total variation regularized deep image prior. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 7715–7719. IEEE, 2019. 12

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3730–3738. IEEE Computer Society, 2015. 7

[19] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 2*, pages 416–425. IEEE Computer Society, 2001. 7

[20] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015. 3

[21] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Com-*

puter Science, pages 262–277. Springer, 2020. 1, 3, 5, 7, 8

[22] Adnan Qayyum, Inaam Ilahi, Fahad Shamshad, Farid Boussaid, Mohammed Bennamoun, and Junaid Qadir. Untrained Neural Network Priors for Inverse Imaging Problems: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022. 1, 2

[23] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11890–11899. Computer Vision Foundation / IEEE, 2020. 3

[24] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, Nov. 1992. 1, 2, 4

[25] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2488–2498, 2018. 3, 15

[26] Bharat Singh, Soham De, Yangmuzi Zhang, Thomas A. Goldstein, and Gavin Taylor. Layer-specific adaptive learning rates for deep networks. In *14th IEEE International Conference on Machine Learning and Applications, ICMLA 2015, Miami, FL, USA, December 9-11, 2015*, pages 364–368. IEEE, 2015. 3

[27] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2846–2855. Computer Vision Foundation / IEEE, 2021. 2

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 15

[29] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9446–9454. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 5, 6, 7, 11

[30] Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun. Early stopping for deep image prior. *CoRR*, abs/2112.06074, 2021. 1, 2, 7

[31] Yuxin Wu and Kaiming He. Group normalization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2018. 15

[32] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3121–3138, 2023. 3, 5

[33] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry L. Schumaker, editors, *Curves and Surfaces - 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers*, volume 6920 of *Lecture Notes in Computer Science*, pages 711–730. Springer, 2010. 6

[34] Kevin Zhang, Mingyang Xie, Maharshi Gor, Yi-Ting Chen, Yvonne Zhou, and Christopher A. Metzler. Metadip: Accelerating deep image prior with meta learning. *CoRR*, abs/2209.08452, 2022. 2, 5, 7

[35] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3592–3602, 2019. 3

[36] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 597–613. Springer, 2016. 3

[37] Zhong Zhuang, Taihui Li, Hengkang Wang, and Ju Sun. Blind image deblurring with unknown kernel size and substantial noise. *CoRR*, abs/2208.09483, 2022. 4