FASTER NEURAL ARCHITECTURE "SEARCH" FOR DEEP IMAGE PRIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep image prior (DIP) is known for leveraging the spectral bias of the convolutional neural network (CNN) towards lower frequencies in various single-image restoration tasks. Such inductive bias has been widely attributed to the network architecture. Existing studies therefore either handcraft the architecture or use automated neural architecture search (NAS). However, there is still a lack of understanding on how the architectural choice corresponds to the image to be restored, leading to an excessively large search space that is both time and computationally-expensive for typical NAS techniques. As a result, the architecture is often searched and fixed for the whole dataset, while the best-performing one could be image-dependent. Moreover, common architecture search requires ground truth supervision, which is often not accessible. In this work, we present a simple yet effective *training-free* approach to estimate the required architecture for *every image* in advance. This is motivated by our empirical findings that the width and depth of a good network prior are correlated with the texture of the image, which can be estimated during pre-processings. Accordingly, the design space is substantially shrunk to a handful of subnetworks within a given large network. The experiments on denoising across different noise levels show that a subnetwork with proper setups could be a more effective network prior than the original network while being highly under-parameterized, making it not critically require early-stopping as with the original large network.

1 INTRODUCTION

Image restoration tasks such as denoising, inpainting and super-resolution are known as ill-posed problems, where priors are often required for regularizing the solution space. Convolutional neural networks (CNNs) have emerged as the tool of choice for such inverse problems (Mao et al., 2016; Zhang et al., 2017a;b; Pathak et al., 2016; Ledig et al., 2017; Dong et al., 2015; Lim et al., 2017; Yu et al., 2019), due to the common belief that they are powerful in learning natural image priors from large-scale datasets. Yet, as an exception, deep image prior (DIP) (Ulyanov et al., 2018) shows that a CNN by itself can regularize the restoration through its architecture along with early-stopping optimization, when fitting a single degraded image without being exposed to external training data. While such *network priors* have been quite competitive across various inverse problems, how to choose the architecture that better represents the image at hand remains an open question.

Early work by Heckel & Hand (2018) suggested that an under-parameterized network could be a good denoising model that represents the image with a concise set of network weights and limited capacity to overfit to noise. Nevertheless, the model requires handcrafting and even under the same parameter budget there exists multiple possibilities, where model setup improper to the images could still lead to noise-fitting or over-smoothing (A.0.2 Fig. 11). Chen et al. (2020) then leveraged a reinforcement learning based controller with PSNR as the reward to automatically search for the ideal architecture for *the whole dataset*. With no prior knowledge about the suitable architectures, the search space ranges from kernel size and up-sampling types to cross-level skip connections, which can be so large (~ 500 models) that the training and evaluation of all candidate architectures is computationally and time costly. This further prohibits image-wise NAS, yielding sub-optimal restoration for certain images. To alleviate the computational burdens, a recent study (Arican et al., 2022) proposed to rank the models at initialization using metrics that measure the discrepancies between the spectral properties of the initially generated network output and those of the corrupted

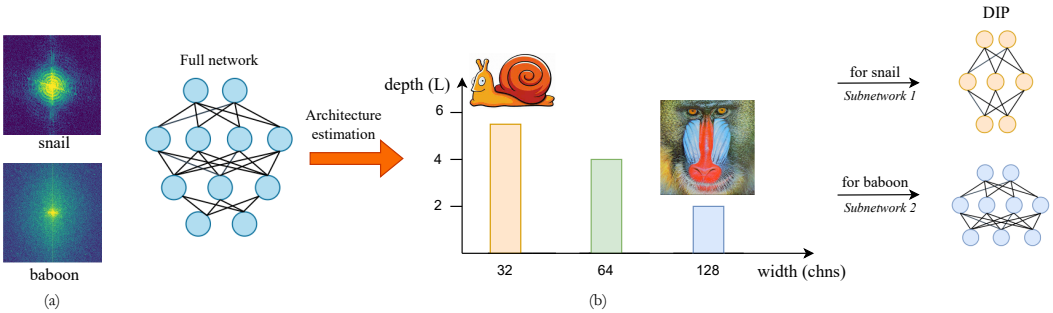


Figure 1: (a) Fourier spectrum of "snail" and "baboon". "Baboon" is a finer-grained image that contains more high frequency components. (b) Overview of the proposed training-free architectural design strategy for *image-specific* Deep Image Prior (DIP) based on the complexity of the image texture.

image. Although they prevent the large-scale model search, due to the imperfect metric, they still need to train 15 models from the candidates pool for each image to determine the top-1 architecture, which substantially prolongs the restoration time.

To improve the efficiency of DIP, *Can we identify and train an (under-parameterized) architecture suitable for each image directly*? This is particularly challenging as there is a large number of different architectures and no ground truth can be used as an explicit supervision. Towards this problem, we study and rethink the architectural design for network priors. We start by noting that (1) there are *not* that many components in a CNN responsible for the denoising effect: the fixed (unlearnt) upsampling plays a primary role; (2) a good architecture should adapt to the texture of the image to balance denoising and preservation of fine details, and surprisingly such balance can be achieved by simply scaling the depth (essentially the number of upsampling layers) and width. Specifically, a finer-grained image tends to require a wider and shallower network, and vice versa. Based on these observations, we restrict the design space to only dozens of subnetworks within a given network, and develop strategies to estimate the optimal width and depth of a subnetwork for each image according to its texture, which can be done as pre-processings without costly evaluations.

We demonstrate that the subnetwork estimated by our simple strategy can better preserve the details while denoising especially on the very fine-grained images, where current methods typically fall short. Moreover, it does not require early-stopping even under a higher noise level, thanks to the under-parameterized nature. To our knowledge, we are the first to report the correlations between the architecture and image texture in the context of DIP and translate these findings into architectural design practices specialized for every image in an efficient manner.

This work makes the following contributions:

- **Fast, zero-cost and image-dependent architectural design** We associate the architectural design for DIP with the complexity (texture, spatial structures) of the images and focus only on the depth and width for a quick and effective design. We show that with proper setups, a highly under-parameterized subnetwork could surpass the original large network.

- **Lightweight, less reliance on early-stopping** The optimal iteration number has been a bottleneck for DIP as it typically varies with images, noise levels and architectures. By contrast, the subnetworks estimated by our strategy require only a fixed iteration number across datasets under both modest and higher noise levels.

- **Base network agnostic** Our empirical strategy is amenable not only to the encoder-decoder architecture as the original DIP, but also to the decoder-only architectures, i.e., Deep Decoder (Heckel & Hand, 2018) and ConvDecoder (Darestani & Heckel, 2021) and improve their performance significantly.

2 RELATED WORK

Deep Image Prior and its Extensions. Under the setting of DIP (Ulyanov et al., 2018), a convolutional neural network (CNN) is randomly-initialized and then optimized to fit a degraded image from a fixed random input vector. Despite not learning from external datasets, the CNN is able to output a recovered image within a certain number of iterations, after which the performance gradually degrades, necessitating early-stopping.

The follow-up works therefore mostly focused on either developing better architectures or studying how to avoid early stopping. Our work pursues both of these two goals by estimating a suitable subnetwork whose capacity aligns with the complexity of the image such that it does not easily over-fit and thus early stopping is not critically required.

– *Architectural choice.* Different from the work by Heckel & Hand (2018) which defined the capacity of an architecture simply by its size (i.e., number of parameters) and used a fixed architecture for all images, we explicitly manipulate the components that are most relevant to the denoising capability of the CNN and customize the architecture for every image. Also, in contrast to the neural architecture search (NAS) based methods (Chen et al., 2020; Ho et al., 2021; Arican et al., 2022), our strategy prevents the exhaustive search by exploiting the observed relationship between the architecture and the image, providing a more interpretable and efficient alternative.

– *Early-stopping.* In order to prevent performance decay due to the learning of too much high-frequency noise, Shi et al. (2022) seeks to constrain the Fourier spectrum of the network by upper bounding the Lipschitz constant of the convolutional layers, and replacing the bilinear up-sampling layers with bandwidth-adjustable Gaussian up-sampling. However, tuning the parameters such as the Lipschitz constant and the sigma of the Gaussian filter is non-trivial for each individual image, especially when the noise level varies. For a comparison, we show that our method attains better results across noise levels without heuristic parameter tuning (see **A.0.1**).

Other extensions include augmenting DIP with explicit reguarlizers such as total variation (Liu et al., 2019) and the Regularization by Denoising (RED) framework (Mataev et al., 2019).

3 PRELIMINARIES AND APPROACH

3.1 BACKGROUND

We resort to Fourier transform for analyzing the up-sampling operations and the characteristics of images of different texture and spatial structures manifested in the frequency domain. For a 2D image $f(x, y)$ of size $M \times N$, the discrete Fourier transform (FT) is defined as:

$$F(k,l) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{xk}{M} + \frac{yl}{N})}, \tag{1}$$

with the spectral coordinates $k = 0, 1, 2, ..., M-1$ and $l = 0, 1, 2, ..., N-1$. As images are signals discretized in space, the highest spatial frequency is constrained by the spatial resolution of the image, given by $f_{nyq} = \sqrt{k^2 + l^2}$. Based on FT, we further compute the power spectral density (PSD), which is the squared magnitudes of the Fourier components: $S(k,l) = \log|F(k,l)|^2$. The spectral power of natural images generally follows an exponential decay with increasing spatial frequencies (Simoncelli & Olshausen, 2001). Coarse and fine textures tend to exhibit different magnitude decay behaviors, as further described in Sec. 3.3.2.

3.2 OBSERVATIONS

When performing denoising on images of different textures under DIP settings, we empirically observed that a deeper model (without skip connections) results in increasingly blurry outputs for finer-grained images, while the more homogeneous image is much less affected (Fig. 2 (b)). This seems to be contrary to our prior belief that the more complicated images require deeper models. On the other hand, when the width is increased, the opposite occurs, where more information unnecessary to the images such as noises are learnt and the simpler image overfits relatively more easily than the more complicated one (Fig. 2 (c)).

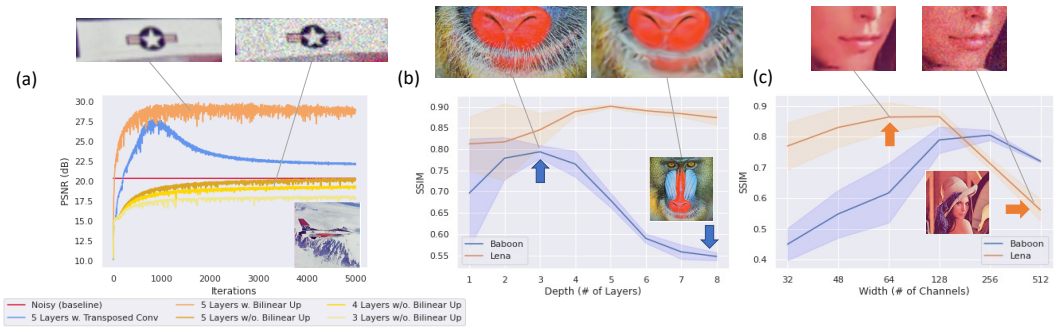


Figure 2: **(a)** Influences of upsampling evaluated on Deep Decoder (Heckel & Hand, 2018). **(b)** Increased depth (with 2×bilinear upsampling layers) over-smooths the image of higher complexity (Baboon), while **(c)** increased width more easily leads to over-fitting to noise for the image of lower complexity (Lena).

By investigating the decoder of the model which corresponds to the reconstruction of the final image, we identify that the fixed upsampling operations inserted in-between the decoder layers is the main driving force behind the denoising effects and we suspect that over-smoothing is caused by the excessive upsampling operations. The results in Fig. 2 (a) suggest the following: both transposed convolutions and fixed (i.e., bilinear) upsampling favor the learning of the low-frequency spatial structures of the image instead of the high frequency noises, which enables denoising, but transposed convolutions introduce high frequencies faster than fixed upsampling, leading to performance drop sooner. Removing the upsampling layers results in loss of most of the denoising effects which cannot be compensated by simply reducing the size of the network (i.e., under-parameterization).

From a frequency perspective, the above findings make sense as bilinear or nearest neighbor interpolation is equivalent to multiplication of the spectrum with a sinc function by the convolution theorem, which suppresses high frequencies. The network therefore acts as a low-pass filter at the early stage of the training, separating the high-frequency noise from natural signals, known as the spectral bias of CNNs (Rahaman et al., 2019; Xu et al., 2019; Chakrabarty & Maji, 2019). Intuitively, the image that is dominated by low frequencies tends to yield good denoising results with the low-pass filtering as most of the energy is concentrated on the center of the spectrum, while the image with a wider bandwidth is more susceptible to losses of high-frequency details. As the number of fixed upsampling layers increases typically along with the depth, the network is more biased towards generating smoother images and thus may not be suitable for fine-grained images. Hence, we argue that fixing the architecture for all images is inherently not optimal and that image texture should be taken into account for cutomized architectural design. Notably, previous theoretical (Heckel & Soltanolkotabi, 2019) have also attributed the denoising effects to the fixed upsampling, but we are the first to associate this effect with the texture of the individual image for more effective denoising.

3.3 Training-free architectural design

Inspired by the observations that depth and width influence images of various texture differently, we localize our focus on these two components and present efficient strategies combined with empirical evidence for estimating them without training. Without loss of generality, we assume that the base network that we estimate the subnetworks from is the same 5-level hourglass architecture with 2× bilinear upsampling inserted in-between each decoder layer as in (Ulyanov et al., 2018), except that skip connections are not included.

3.3.1 Depth estimation

From Fig. 2(b), we observed that a proper depth is more crucial to fine-grained images than to the coarser ones for avoiding over-smoothing. As we have attributed the smoothing effects to the fixed upsampling operations in the decoder as discussed above, we essentially need to determine the number of upsampling layers to keep. An intuition is that the network with a proper depth and

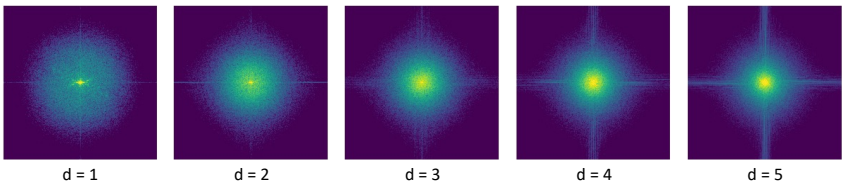


Figure 3: We approximate the influences of the varying depth (with bilinear upsampling layers) on the power spectrum of an image by passing the target image into the subnetworks of DIP at initialization. $d = n$ implies that the subnetwork contains n down- and up-sampling layers. More visualizations in A.0.3.

the fine-grained image to be restored should have similar wide bandwidths, which corresponds to a lower number of down-/upsampling layers. To illustrate this effect, we pass the target image into five independent subnetworks of the DIP with various depths at initialization and compare the PSDs of the generated outputs by computing the bandwidth that contains the 99% of the total energy in the 2D PSD (Fig. 3). As the bilinear interpolation kernels are not learnt, we expect that the spectrum changes can be approximated in this way despite the untrained network weights. The estimation via this way for fine-grained images often leads to $1 \leq d_{est} \leq 2$. In practice, we have found that $d = 2$ almost always achieve the best results while $d = 1$ can lead to under-fitting. Therefore, we keep $d = 2$ throughout the experiments.

3.3.2 Width estimation

A general rule we follow is that a finer-grained image requires wider channels. Each image corresponds to three width choices: $\{32, 64, 128\}$. To more robustly classifying the texture of the images especially when contaminated by noises, we extract features from both the frequency and spatial domains and fuse them to form a single score for each image.

PSD fall-off rate. Coarse texture dominated by large homogeneous spatial structures has its PSD concentrated on low frequencies, which drops off rapidly towards higher frequencies, while finer texture exhibits a more moderate decay. To characterize the decay trend, a straightforward way is to use the ratio of the magnitudes at lower frequencies and the magnitudes at higher frequencies. To do so, we first convert the 2D power spectrum of the image from its Cartesian coordinates to polar coordinates, and compute its 1D representation via azimuthally averaging over θ, defined as $\hat{S}(r) = \frac{1}{2\pi}\int_0^{2\pi} S(r,\theta)d\theta$ with $r = \sqrt{k^2 + l^2}$ and $\theta = \text{atan2}(k, l)$, which represents the mean magnitude of the frequencies with respect to the radial distance r. The averaging of the frequency components along circles also smooths the noise components. For the obtained spectral vector, we drop the DC power and normalize it to $[0, 1]$. For computing the ratio, we avoid the very high frequencies as they are dominated by noises. Specifically, we take the ratio of the low frequency components that comprise $25\% - 50\%$ of the total energy and the high frequency components that comprise $50\% - 75\%$ of the total energy.

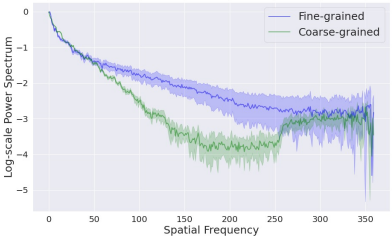


Figure 4: The magnitudes of fine-grained and coarse-grained images (with noises) decay differently.

GLCM features. Texture coarseness can also be characterized by the number of neighboring pixel pairs having the same grey values, which is the basis of the Grey Level Co-occurrence Matrix (GLCM) (Haralick et al., 1973), a classic tool for texture analysis. We derive four kinds of statistics from the GLCM of each image, i.e., dissimilarity ($\sum_i \sum_j c_{ij}|i - j|$), correlation ($-\sum_i \sum_j \frac{(i-u_i)(j-u_j)c_{ij}}{\sigma_i \sigma_j}$), homogeneity ($\sum_i \sum_j \frac{c_{ij}}{1+|i-j|}$) and contrast ($\sum_i \sum_j (i-j)^2 c_{ij}$), where c_{ij} denotes the entry of the GLCM representing the co-occurrences of the pixel values i and j. The co-occurrences are measured by each metric from four angles $\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$, leading to a total of 16 features for each image.

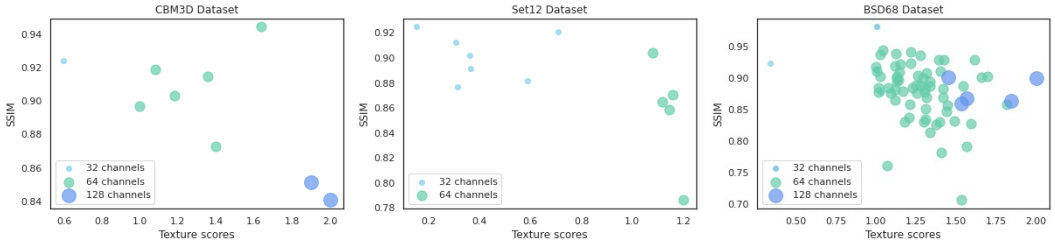


Figure 5: Distribution of the fine- and coarse-grained noisy images scored by our metric. The colors denote the ground truth widths. Our metric enables distinct clusters to form among images.

Image dimensions. Heuristically, we find that the image size itself serves as a simple indicator for narrowing down the width choices. Specifically, the ideal width for smaller-sized (e.g., 256×256) images tends to fall between 32 and 64, and for larger ones (e.g., 512×512) between 64 and 128. This observation turns a multiclass classification problem, which can be more prone to errors in an unsupervised setting as ours, into a simpler binary classification problem. In practice, we simply assign each image with an additional feature $S_{size} \in \{0, 1\}$ indicating its relative size.

Scoring. To score each image by its texture, we first perform PCA on the extracted 1 frequency and 16 spatial features to combine them into a single score S_{pca} normalized to $[0, 1]$, which sufficiently characterizes the most variances among the images. The final score is defined as $S = S_{pca} + S_{size}$.

4 Experiments and Results

Datasets. We conducted the denoising experiments on three standard and publicly-available datasets, 1) CBM3D by Dabov et al. (2007) consisting of 9 colored images, 2) Set12 by Zhang et al. (2017a) consisting of 12 grey-scaled images, and 3) CBSD68 by Roth & Black (2009) consisting of 68 colored images. Additive Gaussian white noise is applied to each image with two noise levels: 25 and 50. The goal is to recover the latent clean image.

Implementation details. We consider three representative architectures commonly-used in DIP settings: 1) DIP (Ulyanov et al., 2018), a 5-level encoder-decoder architecture with strided convolutional layers, $2\times$ bilinear upsampling layers and skip connections. 2) DeepDecoder (Heckel & Hand, 2018), an under-parameterized 5-layer decoder-only architecture with only 1×1 convolutions and bilinear upsampling layers. 3) ConvDecoder (Darestani & Heckel, 2021) a convolutional variant of Deep Decoder with 1×1 convolutions replaced by 3×3 counterparts. All three architectures contain 128 channels per layer. We also compare with two recently-proposed NAS approaches: 4) NAS-DIP (Chen et al., 2020) and 5) ISNAS-DIP (Arican et al., 2022). All compared architectures are trained to minimize $\|y - y_{\text{noisy}}\|_2^2$, where y is the network output and y_{noisy} is the noisy image (full formulation in A.0.5). We use Adam as the optimizer with a constant learning rate of 0.01 for all methods. All other methods are trained for 1200 iterations as in their original settings, and early stopping is applied when required. For our method we use a fixed iteration number 3000 throughout the experiments. Following DIP (Ulyanov et al., 2018), we apply an exponential sliding window to the output image with weight $\gamma = 0.99$ for all compared methods. Peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) are used for quantitative evaluation.

	CBM3D	Set12	BSD68
σ_{25}	0.745	0.728	0.527
σ_{50}	0.745	0.715	0.468

Table 1: Kendall's tau correlation between the complexity of the image texture and the required width.

Accuracy of width estimation. We compare the estimated widths against the ground truth widths which we obained by pre-training all subnetworks with all candidate widths. Fig. 5 shows that for CBM3D dataset and Set12 dataset, our metric well distinguishes the images of various texture with correct widths assigned to all images. For BSD68 dataset, some ambiguities occur on the border between the medium- and the finest-grained images. Tab. 1 shows that the required widths are positively correlated with the texture scores, though a higher noise level induces a slight drop.

Datasets		DIP	Deep Decoder	ConvDecoder	NAS-DIP	ISNAS-DIP	Ours
				$\sigma = 25$			
CBM3D	PSNR	<u>29.15</u>	28.45	28.51	29.07†	29.11†	**30.26**
Dabov et al. (2007)	SSIM	0.864	0.848	0.854	<u>0.865</u>†	0.862†	**0.900**
Set12	PSNR	<u>27.90</u>	27.06†	26.28†	27.04†	26.99†	**28.14**
Zhang et al. (2017a)	SSIM	0.872	0.853†	0.840†	0.832†	**0.885**†	<u>0.884</u>
BSD68	PSNR	26.52	25.50	25.19	<u>28.12</u>†	27.61†	**28.57**
Roth & Black (2009)	SSIM	0.846	0.809	0.793	0.875†	<u>0.875</u>†	**0.888**
				$\sigma = 50$			
CBM3D	PSNR	**26.90**†	25.38†	25.01	26.10†	25.93†	<u>26.13</u>
Dabov et al. (2007)	SSIM	<u>0.826</u>†	0.771†	0.769	0.791†	0.799†	**0.833**
Set12	PSNR	**25.11**†	23.10†	22.72	<u>25.08</u>†	24.19†	24.59
Zhang et al. (2017a)	SSIM	0.805†	0.746†	0.706	**0.810**†	0.767†	<u>0.805</u>
BSD68	PSNR	**24.80**†	23.66†	24.36	<u>24.62</u>†	22.97†	24.17
Roth & Black (2009)	SSIM	<u>0.780</u>†	0.731†	0.767	**0.782**†	0.763†	0.774
# Params (Millions)		2.3M	**0.1M**	0.89M	4.4M	*Varied*	**0.05M∼0.92M**

Table 2: **Quantitative results on denoising experiments**. σ denotes the noise level. † denotes that early-stopping is required and applied (where we have access to the ground truth for determining the stopping iteration). Our method adopts a fixed iteration number throughout the experiments. A comparison with another line of work focused on automatic stopping criterion (Shi et al., 2022) is provided in A.0.1 The highest score is in **bold**, and the second highest is <u>underlined</u>.

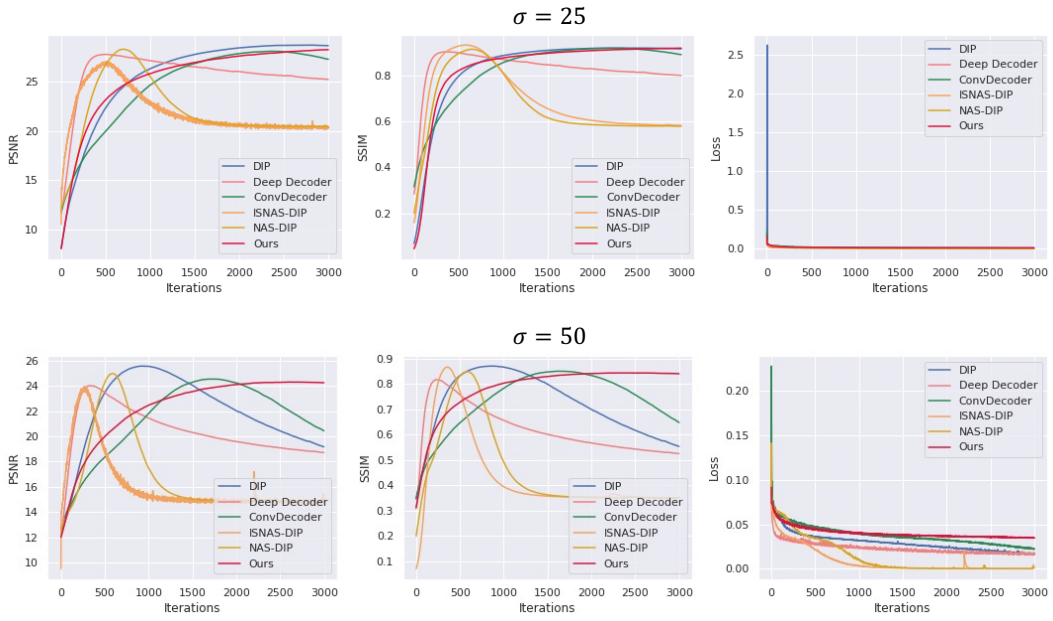


Figure 6: Metric curves of all compared methods across noise levels for an image from Set12. All previous methods suffer from over-fitting especially when the noise level is higher, including the under-parameterized ones (i.e., ConvDecoder, Deep Decoder). The optimal iteration number varies with the architectures, noise levels and images, which can be hard to determine in practice without accessing to the ground truth.

Method comparisons. We compare the subnetworks estimated from DIP with our architectural design strategy to the original DIP architecture and also to other handcrafted and automated methods. Tab. 2 shows that under a moderate noise level, our method performs best on all three datasets. This is largely because the hard cases in the datasets such as those with abundant fine details (Fig. 7 Baboon), where current methods tend to over-smooth, are handled well with our properly setup

depth and width, indicating the importance of the texture-based and image-specific architectural design (see A.0.2 Fig. 10 for another example). ISNAS-DIP is also developed for image-specific architecture search, but it does not explicitly take the texture into account and over-smooths the

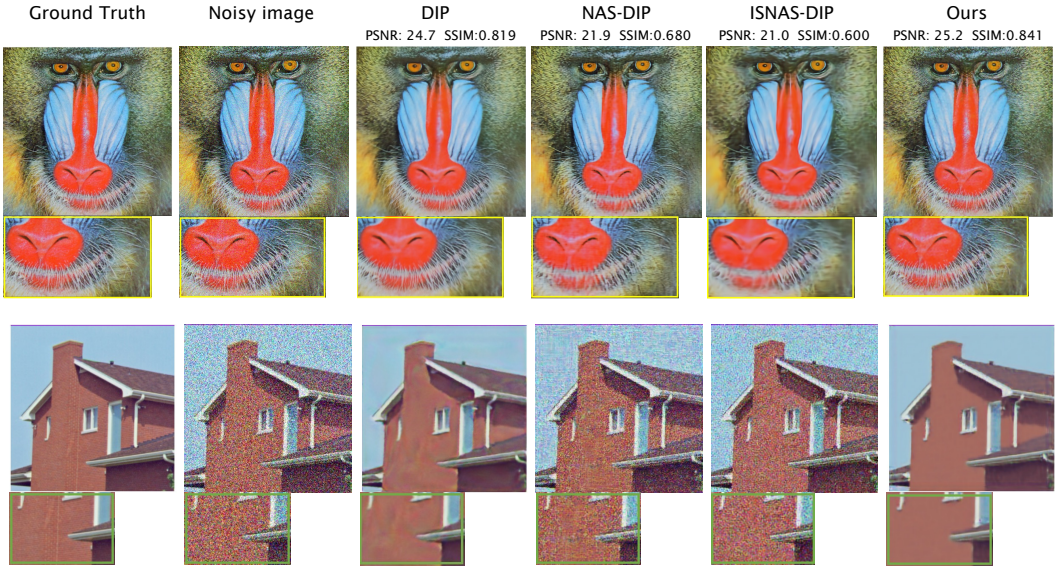


Figure 7: Qualitative examples of image denoising under the noise level 25 (top row) and 50 (bottom row) respectively. The architectures used in our approach for Baboon and House are two different subnetworks of DIP. Other methods tend to over-smooth the image. Our method well preserves the fine details while denoising and does not require early stopping even under a higher noise level. For space considerations, visualizations for Deep Decoder and ConvDecoder are provided in A.0.2.

image most on the very fine-grained images. NAS-DIP heavily relies on early-stopping and does not show significant advantages over DIP considering its resource-intensive searching. Our method outperforms DIP especially on BSD68, where images with various degrees of texture are present, suggesting that a more flexibly-designed architecture for each image is desired. Moreover, our estimated subnetworks contain only $2\% \sim 40\%$ of the total parameters of the original DIP. Such under-parameterized nature makes them robust even under a higher noise level (Fig. 6), which is a more challenging setting as the noise can now be more dominant at lower frequencies and thus separating them from the useful low frequencies becomes more difficult. Our method thus does not require much parameter tuning as other methods do. Note that DIP and Deep Decoder necessitate early stopping in this case. ConvDecoder relies less on early-stopping at the expense of performance.

On the influences of under-parameterization. Deep Decoder is a highly under-parameterized architectures with the number of parameters much less than the dimensions of the image it represents and ConvDecoder is its convolutional variant, while they either fail to remove the noises completely or over-smooth the image (A.0.2 Fig. 11). Fig. 6 also shows that Deep Decoder does not exhibit stronger resistance to overfitting than ConvDecoder despite being lighter. This is consistent with our observations in Fig. 2 (a) that the spatial convolutional layers (e.g., transposed convolutions) have certain smoothing effects that slow down the learning of the high frequencies while the 1×1 convolutions do not, regardless of heavy- or under-parameterization. In this context, Deep Decoder contains only the bilinear upsampling layers for effective denoising while ConvDecoder contains both, which makes it a stronger denoiser and even over-smooth the images. On the other hand, the subnetworks estimated by our method can effectively preserve the details while being lightweight. These findings suggest that without proper model setups, under-parameterization itself can neither ensure good denoising performance, nor remove the need for early-stopping.

On improving decoder-only architectures. To evaluate the generalizability of our strategy, we integrate it into Deep Decoder and ConvDecoder. Quantitative results are summarized in Tab. 3. We have observed that Deep Decoder tends to overfit to noises more easily on simpler images (Fig. 11) while ConvDecoder constantly over-smooths the images (Fig. 8,Fig. 11). Fig. 8 and Fig. 12 show

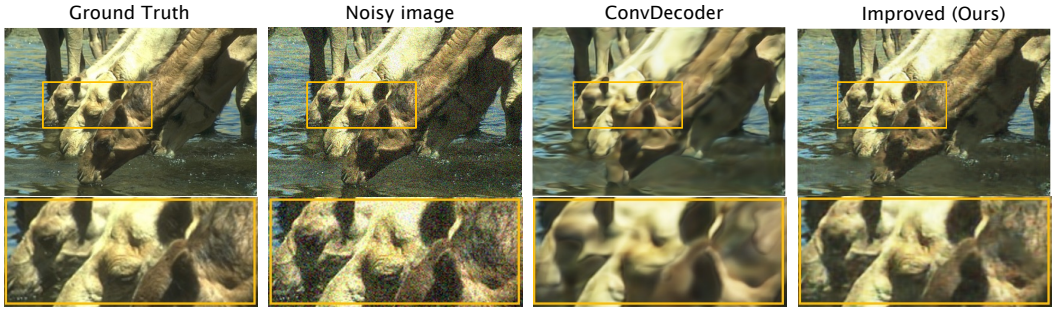


Figure 8: Qualitative example of the improvement on ConvDecoder(Darestani & Heckel, 2021) with our approach. In this case, three upsampling layers were removed from the original ConvDecoder architecture with the width down-scaled, leading to a lighter subnetwork with better performance.

Datasets		Deep Decoder		ConvDecoder	
		Before	After	Before	After
		$\sigma = 25$			
CBM3D	PSNR	28.45	28.71↑	28.51	28.74↑
	SSIM	0.848	**0.860**↑	0.854	**0.873**↑
Set12	PSNR	26.00	26.41↑	25.79	**26.98**↑
	SSIM	0.789	**0.825**↑	0.786	**0.854**↑
BSD68	PSNR	25.50	25.56↑	25.19	**28.29**↑
	SSIM	0.809	0.802	0.793	**0.877**↑
		$\sigma = 50$			
CBM3D	PSNR	25.22	25.62↑	25.01	24.15
	SSIM	0.764	**0.790**↑	0.769	**0.784**↑
Set12	PSNR	22.45	23.01↑	22.72	23.01↑
	SSIM	0.649	**0.687**↑	0.706	**0.742**↑
BSD68	PSNR	23.52	23.60↑	24.36	24.12
	SSIM	0.725	**0.740**↑	0.767	0.755
# Params (Millions)		0.1M	0.006M∼0.1M	0.89M	0.06M∼ 0.89M

Table 3: Integrated our strategy into decoder-only architectures. Performance of all methods at the 3000th iteration is reported.

that our strategy ameliorates these issues. Note that we only apply width scaling on Deep Decoder as it heavily relies on upsampling for denoising as discussed above. For ConvDecoder, we reduce the upsampling operations while preserve the convolutional layers as it is already under-parameterized. The results show that our strategy significantly improves ConvDecoder on all datasets especially under the noise level of 25. While at the noise level of 50, the improvement becomes moderate. This is consistent with the results in Tab.2. We speculate that the down-scaled width impairs the ability to preserve fine details under a higher noise level, though it effectively avoids noise-fitting, an inherent trade-off in denoising through spectral bias. Moreover, the improvement due to the removal of the excessive upsampling layers without changing the network depth supports our hypothesis that the upsampling operation plays a major role in controlling the extent of smoothing. For Deep Decoder, the improvement is also appreciable even with only the width scaling, further demonstrating the strong correlations between the complexity of the image texture and the proper width.

5 Conclusions

In this paper, we present a surprisingly efficient solution to the two open challenges of DIP regarding architectural choice and early-stopping. By observing that increasing the network depth tends to over-smooth the fine-grained images, we hypothesize that the excessive fixed upsampling operation is the main cause and propose to choose the number of required upsampling layers along with the network width based on the complexity of the image texture. With simple architectural changes, a high-performing under-parameterized architecture could surpass the much larger ones in denoising while well-preserves the details with reduced risks of over-fitting and thus can endure a higher noise level without requiring early-stopping. This greatly alleviates the burdens of parameter tuning in practice when ground true images are not available. In contrast to most existing resource-intensive architecture search methods for DIP, the proposed method only manipulates the width and depth of the network for each image, which is zero-cost and training free. Complementing previous studies (Heckel & Hand, 2018; Heckel & Soltanolkotabi, 2019), the presented findings provide insights into design of more effective under-parameterized architectures. We hope that this study could encourage efficient architectural design for DIP and image synthesis in general.

REFERENCES

Metin Ersin Arican, Ozgur Kara, Gustav Bredell, and Ender Konukoglu. Isnas-dip: Image-specific neural architecture search for deep image prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1960–1968, 2022.

Prithvijit Chakrabarty and Subhransu Maji. The spectral bias of the deep image prior. *arXiv preprint arXiv:1912.08905*, 2019.

Yun-Chun Chen, Chen Gao, Esther Robb, and Jia-Bin Huang. Nas-dip: Learning deep image prior with neural architecture search. In *European Conference on Computer Vision*, pp. 442–459. Springer, 2020.

Kostadin Dabov, Alessandro Foi, and Karen Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *2007 15th European Signal Processing Conference*, pp. 145–149. IEEE, 2007.

Mohammad Zalbagi Darestani and Reinhard Heckel. Accelerated mri with un-trained neural networks. *IEEE Transactions on Computational Imaging*, 7:724–733, 2021.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2): 295–307, 2015.

Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7890–7899, 2020.

Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *arXiv preprint arXiv:1810.03982*, 2018.

Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. *arXiv preprint arXiv:1910.14634*, 2019.

Kary Ho, Andrew Gilbert, Hailin Jin, and John Collomosse. Neural architecture search for deep image prior. *Computers & graphics*, 98:188–196, 2021.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7715–7719. Ieee, 2019.

Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29, 2016.

Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Stefan Roth and Michael J Black. Fields of experts. *International Journal of Computer Vision*, 82 (2):205–229, 2009.

Zenglin Shi, Pascal Mettes, Subhransu Maji, and Cees GM Snoek. On measuring and controlling the spectral bias of the deep image prior. *International Journal of Computer Vision*, 130(4):885–908, 2022.

Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454, 2018.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4471–4480, 2019.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26 (7):3142–3155, 2017a.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3929–3938, 2017b.

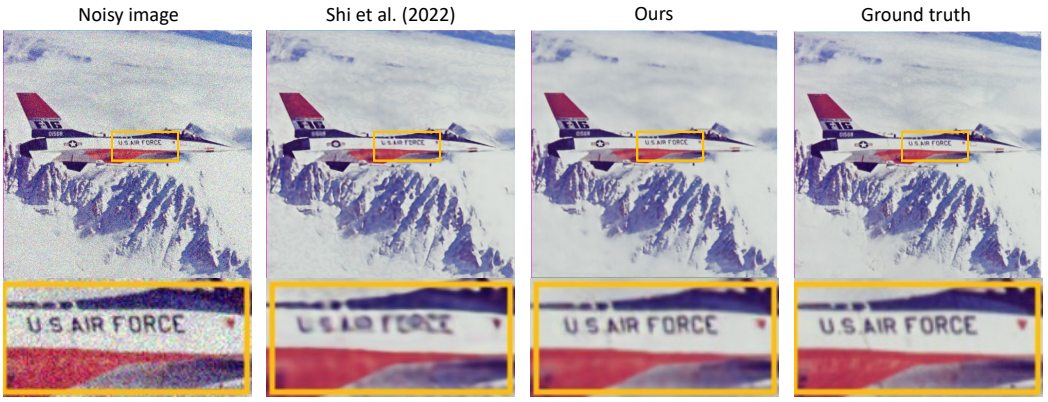
A Appendix

A.0.1 Additional comparisons

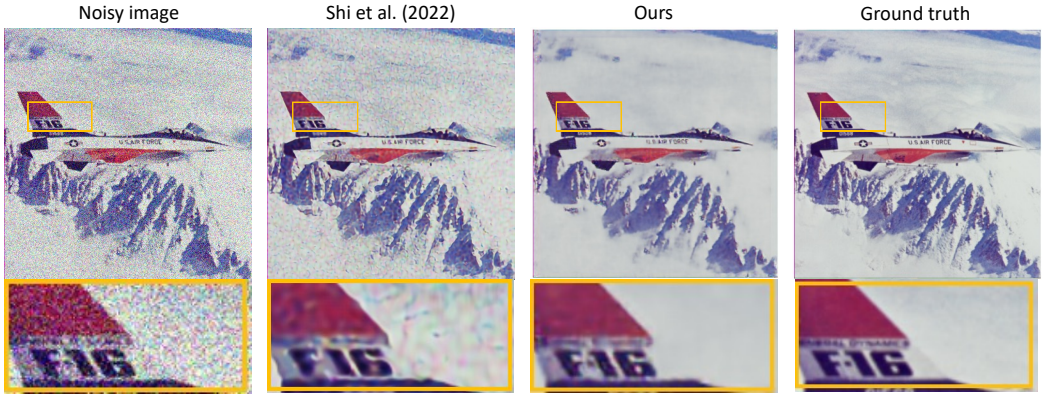
While we focus on architectural design in the main text, our method is also advantageous in avoiding the early-stopping oracle. Here we compare our method with the recently proposed method (Shi et al., 2022) that explicitly removes the need for early-stopping by controlling the spectral bias of the network. Qualitative and quantitative results on CBM3D dataset (Dabov et al., 2007) under modest to larger noise levels are provided below. For parameter choices, we followed Shi et al. (2022) and chose the value of lambda in Lipschitz normalization to be 1.8, and the sigma of the Gaussian upsampling varied with the layer, i.e., $\{0.1, 0.1, 0.1, 0.5, 0.5\}$. The optimization stops automatically when the criterion is met ($\sim$ 4500 iterations). For our method, we used a fixed iteration number of 3000 for both noise levels. From the results, it can be seem that the method by Shi et al. (2022) prevents early-stopping at the expense of performance, especially when the noise level is higher.

Method	$\sigma = 25$		$\sigma = 50$	
	PSNR	SSIM	PSNR	SSIM
Shi et al. (2022)	28.45	0.851	24.28	0.706
Ours	**30.10**	**0.903**	**25.91**	**0.839**

Table 4: Quantitative results on CBM3D dataset with varying noise levels.



(a) $\sigma = 25$



(b) $\sigma = 50$

Figure 9: Qualitative comparison across modest to larger noise levels.

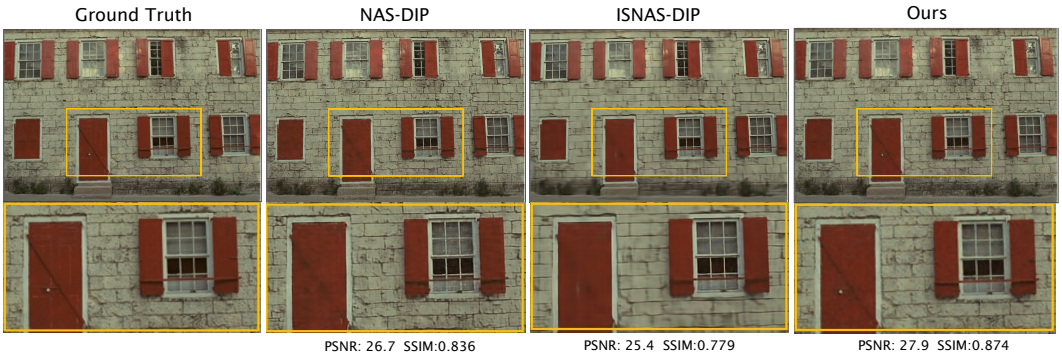


Figure 10: Another hard case in CBM3D dataset. Over-smoothing can be observed with NAS-DIP (Chen et al., 2020) and ISNAS-DIP (Arican et al., 2022). Our method well preserves the details. Evaluated under the noise level 25.

A.0.2 More qualitative examples

A.0.3 More visualizations on the influences of Up-sampling

To supplement Fig. 3, we investigate whether the change in the power spectcrum is mainly caused by the fixed upsampling layers. To do so, we removed the upsampling layers from all the subnetworks.

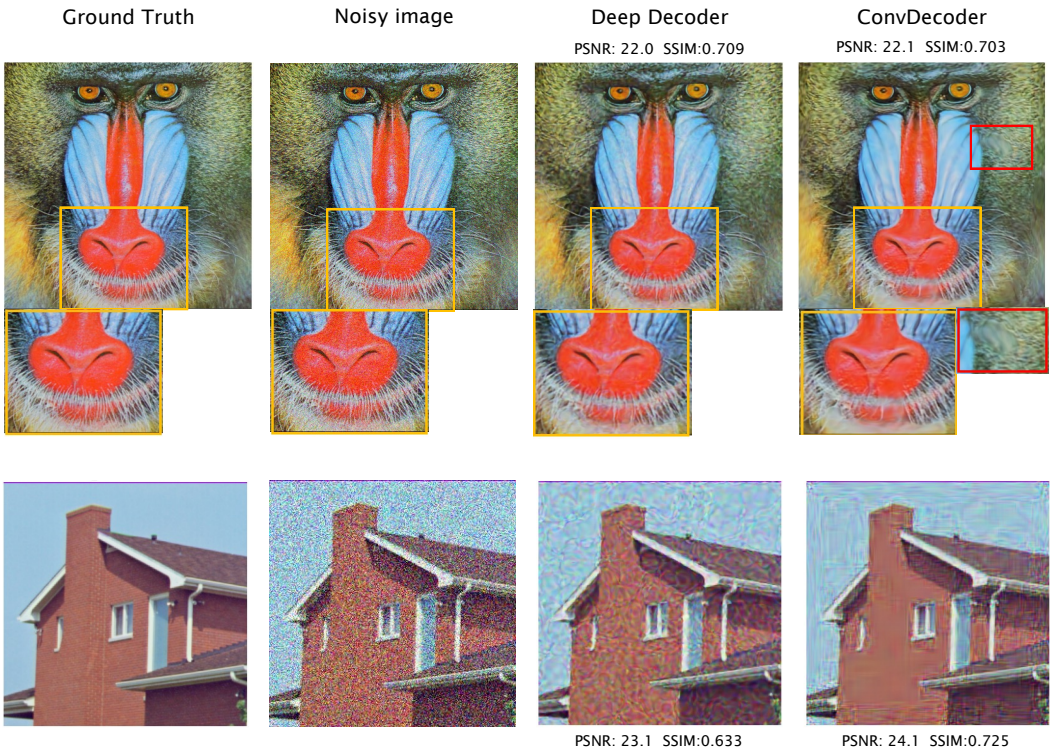


Figure 11: Qualitative example of Deep decoder and ConvDecoder under noise level 25 (top row) and 50 (bottom row) respectively. In both cases, Deep Decoder tends to over-fit more easily to noises, despite its underparameterized nature, while ConvDecoder tends to over-smooth the image.

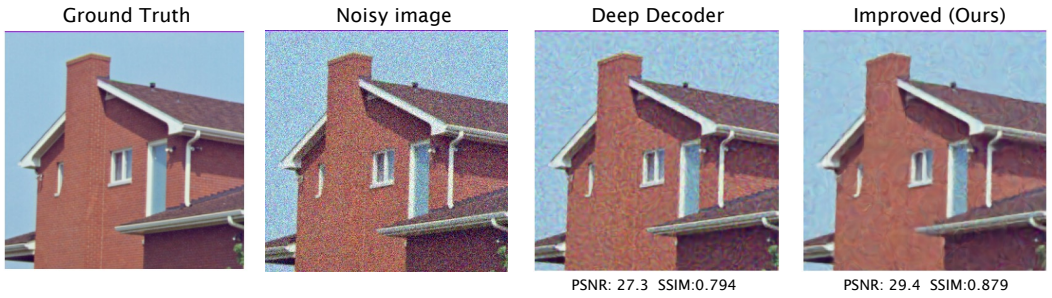


Figure 12: Qualitative example of the improvement on DeepDecoder (Heckel & Hand, 2018) with our approach. Evaluated under the noise level 25.

As this would not be feasible with the encoder-decoder architecture that contains downsampling, we conducted the experiments on a decoder-only architecture, i.e., ConvDecoder (Darestani & Heckel, 2021). Fig. 13(a) shows that the convolutional layers with randomly-initialized weights do not induce much change. We further show an example of passing a coarser-grained image or noise instead as compared to the one in the main text (in Fig. 3), shown in Fig. 14 and Fig. 15 respectively, where change can be seen.

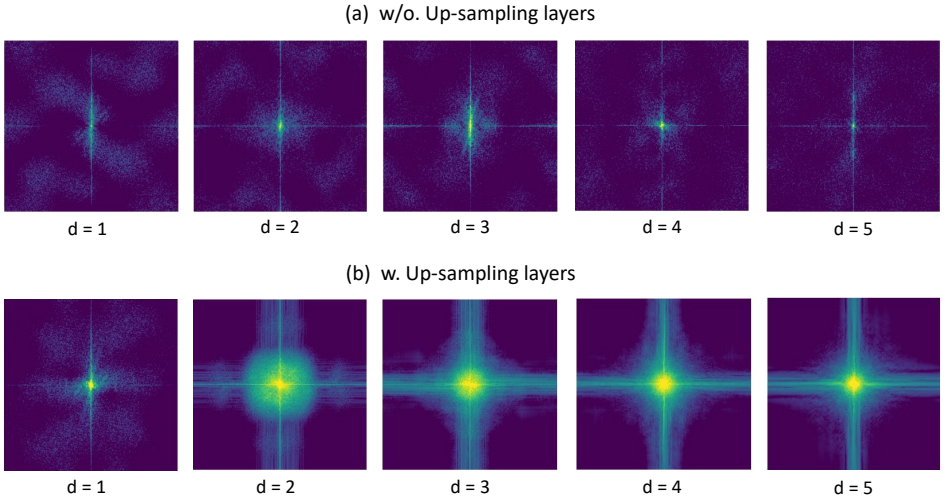


Figure 13: Examples of how the depth influences the power spectrum of the generated image with and without the fixed up-sampling layers, evaluated on a 5-layer **decoder-only** network by passing the target image into the subnetworks of different depths at initialization.

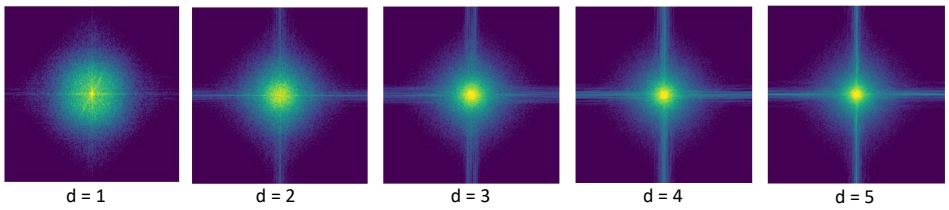


Figure 14: Here we passed a coarser-grained image into the subnetworks at initialization, as a direct comparison with Fig. 3. Evaluated on **DIP**.

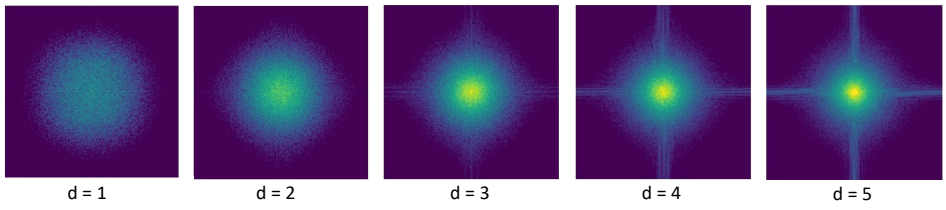


Figure 15: To quantify the spectrum change of each subnetwork which is independent of the image to be restored, we passed the noise instead of the noisy image into the subnetworks at initialization, evaluated on **DIP** as a direct comparison with Fig. 3.

A.0.4 FOURIER ANALYSIS FOR UPSAMPLING

For simplicity, we consider the case of a 1D signal $x(n)$ and its discrete Fourier Transform $X(k) = \sum_{n=0}^{N-1} x(n) \exp\left(\frac{-i2\pi}{N} kn\right)$. When increasing its spatial resolution by factor 2, we have:

$$\begin{aligned} X^{up}(\hat{k}) &= \sum_{n=0}^{2N-1} x^{up}(n) \exp \frac{-i2\pi}{2N} \hat{k}n \\ &= \sum_{n=0}^{N-1} x'(2n) \exp \frac{-i2\pi}{2N} (2n)\hat{k} + \sum_{n=0}^{N-1} x'(2n+1) \exp \frac{-i2\pi}{2N} 2(n+1)\hat{k} \\ &\text{for } \hat{k} = 0, ..., 2N-1. \end{aligned}$$

In the case of transposed convolution where zeros are inserted in the upsampled sequence, $x'(2n) = x(n)$ and $x'(2n+1) = 0$ for $n = 0, ..., N-1$. Hence, for $\hat{k} < N$, $X^{up}(\hat{k}) = X(k)$.

For $\hat{k} \geq N$, let $k' = \hat{k} - N, k' = 0, ..., N-1$:

$$\begin{aligned} X^{up}(\hat{k}) &= \sum_{n=0}^{N-1} x(n) \exp(\frac{-i2\pi}{2N}(k'+N)2n) \\ &= \sum_{N-1}{}_{n=0} (x(n) \exp(\frac{-i2\pi}{N} nk' - i2n\pi)) = X(k'), \end{aligned}$$

due to the fact that the exponential function is periodic with an imaginary period. Therefore, zero-insertion will create high frequency replica of the original low-frequency contents $X(k)$ at $[N, 2N-1]$, in addition to the original copy at $[0, N]$. When nearest neighbor or bilinear upsampling is used, $x'(2n+1)$ will be the average of the adjacent values of $x(n)$, and this is equivalent to multiplication of $X^{up}(\hat{k})$ with a sinc function by convolution theorem that corresponds to low-pass filtering (Gonzalez, 2009; Durall et al., 2020).

A.0.5 TRAINING OBJECTIVE

Consider a general linear inverse problem $y = \mathbf{A}\mathbf{x} + \epsilon$, where $\mathbf{A}$ is any known degradation matrix and in the case of denoising $\mathbf{A} = \mathbf{I}$, and ϵ is assumed to be Additive White Gaussian Noise (AWGN) of variance σ^2. The recovered image $\mathbf{x}$ can be obtained from the degraded image $\mathbf{y}$ by solving

$$\min_{\mathbf{x}} \mathcal{L}(\mathbf{y}; \mathbf{A}\mathbf{x}) + \lambda \mathcal{R}(\mathbf{x}), \tag{2}$$

where $\mathcal{R}(\cdot)$ denotes the prior for regularization. DIP, on the other hand, removes the explicit regularizer and parameterizes the image $\mathbf{x}$ via a neural network $\mathbf{G}_\theta(\mathbf{z})$ given a fixed noise vector $\mathbf{z}$:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\mathbf{y}; \mathbf{A}\mathbf{G}_\theta(\mathbf{z})), \quad \mathbf{x}^* = \mathbf{G}_{\theta^*}(\mathbf{z}). \tag{3}$$

This parameterization allows novel priors to be designed based on the implicit regularization imposed by the architecture of the network $\mathbf{G}_\theta(\cdot)$, instead of in the image space as captured by $\mathcal{R}(\cdot)$.