International Journal of Computer Vision (2022) 130:885–908
https://doi.org/10.1007/s11263-021-01572-7

On Measuring and Controlling the Spectral Bias of the Deep Image Prior

Zenglin Shi[1] · Pascal Mettes[1] · Subhransu Maji[2] · Cees G. M. Snoek[1]

Received: 2 July 2021 / Accepted: 20 December 2021 / Published online: 11 February 2022

Abstract
The deep image prior showed that a randomly initialized network with a suitable architecture can be trained to solve inverse imaging problems by simply optimizing it's parameters to reconstruct a single degraded image. However, it suffers from two practical limitations. First, it remains unclear how to control the prior beyond the choice of the network architecture. Second, training requires an oracle stopping criterion as during the optimization the performance degrades after reaching an optimum value. To address these challenges we introduce a frequency-band correspondence measure to characterize the spectral bias of the deep image prior, where low-frequency image signals are learned faster and better than high-frequency counterparts. Based on our observations, we propose techniques to prevent the eventual performance degradation and accelerate convergence. We introduce a Lipschitz-controlled convolution layer and a Gaussian-controlled upsampling layer as plug-in replacements for layers used in the deep architectures. The experiments show that with these changes the performance does not degrade during optimization, relieving us from the need for an oracle stopping criterion. We further outline a stopping criterion to avoid superfluous computation. Finally, we show that our approach obtains favorable results compared to current approaches across various denoising, deblocking, inpainting, super-resolution and detail enhancement tasks. Code is available at https://github.com/shizenglin/Measure-and-Control-Spectral-Bias.

Keywords Spectral bias · Deep image prior

1 Introduction

This paper considers the problem of inverse imaging, where the task is to recover the original image from the one that is degraded due to noise, blur, down-sampling and other hardships (Bertero and Boccacci 1998). This problem is ill-posed, as a degraded image may correspond to several original images. Hence, reconstructing a unique solution that fits the degraded image is difficult, or impossible even, without some prior knowledge about the image or the degradation (Engl et al. 1996).

The classical computer vision approaches to inverse imaging minimize a regularized cost function to incorporate some prior knowledge into the solution, *e.g.*, (Hahn et al. 2011; Dong et al. 2015c; Arias et al. 2011; Lin et al. 2008). Despite their excellent results, it remains difficult to handcraft an appropriate regularizer and choose a suitable regularisation parameter for a given application because expert knowledge is often required (Ribes and Schmitt 2008; Jin et al. 2017). Rather than providing the priors as input, deep neural networks offer the ability to learn image priors from numerous image samples, *e.g.*, (McCann et al. 2017; Lucas et al. 2018; Arridge et al. 2019). By doing so, the image priors are gradually encoded into network parameters during training and reused in the inference phase. Despite its promise, the dependence on image pairs seen during training may result in poor generalization of the learned priors Zhang et al. (2017, 2018).

Contrary to the belief that learning on numerous image samples is necessary to obtain useful image priors, Ulyanov

Communicated by Gang Hua.

✉ Zenglin Shi
z.shi@uva.nl

Pascal Mettes
P.S.M.Mettes@uva.nl

Subhransu Maji
smaji@cs.umass.edu

Cees G. M. Snoek
cgmsnoek@uva.nl

[1] University of Amsterdam, Amsterdam, Netherlands

[2] University of Massachusetts, Amherst, USA

et al. (2018, 2020) show that the architecture of a generator network itself contains an inductive bias independent of learning, where a *deep image prior* can be implicitly captured by a particular network architecture like an encoder-decoder. To leverage the deep image prior for solving inverse imaging problems, a suitably designed network is optimized, starting from a random initialization and a random input, on just a single degraded image through gradient descent. The network is able to output a well-restored image, when its optimization is stopped at the right time, with an early-stopping oracle. The literature studying the deep image prior mostly focuses on designing network architectures (Heckel and Hand 2019; Cheng et al. 2019; Chen et al. 2020b; Ho et al. 2020). However, it remains unclear how to control the deep image prior beyond the choice of the network architecture and prevent performance degradation when an oracle to stop the optimization at peak performance is unavailable. In this paper, we study the deep image prior from a complementary perspective to address these problems.

As our first contribution, we study the deep image prior through measuring its spectral bias (Sect. 3). We find that both the networks of the original deep image prior (Ulyanov et al. 2018, 2020) and its variants (Heckel and Hand 2019; Cheng et al. 2019) exhibit a spectral bias during optimization, where the low frequency components of the target images are learned better and faster than the high-frequency components. We believe that the spectral bias leads the networks to capture deep image priors during optimization, beyond the choice of the architecture, since natural images are well approximated by low-frequency components according to the power spectrum (Simoncelli and Olshausen 2001). We measure the spectral bias with a new Frequency-Band Correspondence metric and pinpoint why the performance of the deep image prior gradually degrades after reaching a peak during the optimization.

We observe that deep image prior performance degrades when high-frequency noise is learned beyond a certain level, which could affect the high-frequency image details. As our second contribution, we therefore propose to prevent performance degradation by restricting the ability of the network to fit high-frequency noise (Sect. 4). We bound the layers of our network with Lipschitz regularization and introduce a Lipschitz-variant of batch normalization to accelerate and stabilize the optimization. We also observe that widely used upsampling methods, like bilinear upsampling, over-smooth, which introduces a bias towards lower frequencies. This slows down the learning of the desired higher frequencies, delaying optimization convergence. Therefore, we propose an upsampling method which allows controlling the amount of smoothing and is capable of balancing performance and convergence. Besides these two methods for controlling spectral bias, we further introduce a simple automatic stopping criterion to avoid superfluous computation.

Lastly, we demonstrate the effectiveness of our method on four inverse imaging applications and one image enhancement application: image denoising, JPEG image deblocking, image inpainting, image super-resolution and image detail enhancement (Sect. 5). The experiments show that our method no longer suffers from eventual performance degradation during optimization, relieving us from the need for an oracle criterion to stop early. The automatic stopping criterion avoids superfluous computation. Our method also obtains favorable restoration and enhancement results compared to current approaches, across all tasks.

2 Related Work

2.1 Inverse Problems in Imaging

An inverse problem in imaging is the task of recovering an unknown image $x^* \in \boldsymbol{X}$ from its noisy measurements $y \in \boldsymbol{Y}$, where $y = \mathcal{A}(x^*) + e$. Here $e \in \boldsymbol{Y}$ denotes some noise in the measurements. The mapping $\mathcal{A} : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ denotes the forward operator, which could represent various inverse problems, such as an identity operator for image denoising, convolution operators for image deblurring, filtered subsampling operators for super-resolution, *etc.* Since the operator $\mathcal{A}$ has a non-trivial null space, these inverse problems are often ill-posed. Meaning that the solution is unstable with respect to the measurements, or there are several possible solutions that are consistent with the measurements (Bertero and Boccacci 1998). To solve these ill-posed inverse problems, we review the classical knowledge-driven approaches and the recent data-driven approaches with deep neural networks.

The classical knowledge-driven approaches assume some prior knowledge about the image x^*, such as smoothness (Titterington 1985; Katsaggelos 1989) or sparsity (Daubechies et al. 2004; Elad et al. 2010). These approaches typically aim to find a solution that fits well with the measurements y and is consistent with the assumed prior knowledge. To do so, an optimization criterion is used, such as the minimization of the l_2 error norm $||y - \mathcal{A}(x^*)||^2$. Then, prior knowledge is incorporated into the solution process through regularization. Specifically, Rudin et al. (1992) leveraged the fact that in natural images nearby pixels tend to have similar values, and proposed a denoising model with the total variation regularization, which promotes smoothness while preserving edges in images. Based on the finding that natural images can be generally coded by structural primitives such as edges and line segments (Olshausen and Field 1996), sparse representation-based regularization models, *e.g.*, (Elad et al. 2010; Daubechies et al. 2004; Portilla 2009), have been successfully used in image deconvolution tasks. A natural image often has many repetitive local patterns, and thus a local image patch always has many similar patches across the image (Efros and Leung 1999). This non-local self-similarity prior was later employed in many inverse imaging problems such as image denoising (Dabov et al. 2007), image deblurring (Kindermann et al. 2005) and super-resolution (Protter et al. 2008). Later, Mairal et al. (2009) proposed non-local sparse regularization models which combine the local sparsity and the non-local self-similarity into a unified framework, where the similar image patches are simultaneously coded to improve the robustness of the inverse reconstruction. Despite their excellent results, a downside of these approaches is that their handcrafted regularization only captures a fraction of the prior knowledge about the image, limiting the inverse imaging ability of their models (Ribes and Schmitt 2008; Jin et al. 2017).

Data-driven approaches leverage large collections of training data to directly compute regularized reconstructions with deep neural networks. The central idea is to create a paired dataset of ground truth images x and corresponding measurements y, which can be done by simulating (or physically implementing) the forward operator $\mathcal{A}$ on clean data. Subsequently, one can train a network to learn a direct mapping from measurements y to the ground truth images x. Most approaches have focused on designing a proper network architecture to learn a high-performing mapping. For example, Dong et al. (2015b) learned a convolutional neural network for image super-resolution, and Jain and Seung (2008) learned a convolutional neural network for image denoising. Mao et al. (2016) demonstrated convolution neural networks with encoder-decoder architectures perform better for restoring degraded images. Zhang et al. (2017) proposed to use the convolution neural networks with residual blocks and skip connections to further improve image super-resolution and denoising performance. Ledig et al. (2017) proposed a generative adversarial network for image super-resolution to recover the finer texture details. Li et al. (2018) proposed a computationally efficient frequency domain deep network for image super-resolution. Despite their excellent results, these approaches are sensitive to changes or uncertainty to the forward operator $\mathcal{A}$. For image denoising, for example, a specific network needs to be trained for each considered noise level. To remedy this issue, Lefkimmiatis (2018) proposed a universal denoising network with non-local filtering layers, which is able to handle a wide range of noise levels using a single set of learned parameters. Recently, Chen et al. (2020a) proposed a plugin module, which can be inserted into any backbone networks. This plugin allows the once trained network to be used for multiple forward operators in various image processing tasks, including image smoothing, image denoising, image deblocking, image enhancement and neural style transfer. Wan et al. (2020) proposed a triplet domain translation network for restoring old photos, in which multiple degradations exist and are mixed. Such supervised approaches typically perform very well but rely on a paired dataset of ground truth images and their measurements, which may not be available. In this work, we consider the unsupervised inverse imaging approach with a deep image prior.

2.2 Deep Image Prior

The deep image prior, introduced by Ulyanov et al. (2018, 2020), revealed the remarkable ability of untrained convolution neural networks to solve challenging inverse problems by optimizing on just a single degraded image. Let $f_\theta : \boldsymbol{Z} \rightarrow \boldsymbol{Y}$ denote a convolutional neural network parameterized by $\theta \in \Theta$, which transforms a tensor/vector $z \in \boldsymbol{Z}$ to a degraded image $y \in \boldsymbol{Y}$. Without training, the network f_θ has no knowledge about high-level semantic concepts such as the categories of objects in the images. However, the deep image prior found that the network does contain knowledge about the low-level structure of natural images. This prior knowledge is sufficient to model the conditional image distribution $p(x^*|y_0)$. Here, the unknown image x^* has to be determined given a measurement y_0, which allows solving inverse problems in imaging. Specifically, we consider energy minimization problems of the type, $\theta^* = \arg\min_\theta E(f_\theta(z); y_0)$ where $E(f_\theta(z); y_0)$ is a task-dependent data term. For inverse imaging problems, y_0 is a noisy, low-resolution, compressed, or occluded image. The minimizer θ^* is obtained using an optimizer such as gradient descent, starting from a random initialization of the parameters. Given a minimizer θ^* obtained by N steps of gradient descent, we obtain a restoration result by $y^*{=}f_{\theta^*}(z)$. Competitive performance is even feasible when stopping the network optimization with an early-stopping oracle.

The deep image prior has inspired many to investigate how to expand its applications (Gandelsman et al. 2019; Kattamis et al. 2019; Rasti et al. 2021; Vu et al. 2021; Dai et al. 2020), how to improve its performance (Mataev et al. 2019; Chen et al. 2020b; Liu et al. 2019; Asim et al. 2019; Zukerman et al. 2020), how to understand its workings (Ulyanov et al. 2018, 2020; Cheng et al. 2019; Heckel and Soltanolkotabi 2020), and how to avoid its early-stopping oracle (Cheng et al. 2019; Heckel and Hand 2019).

Liu et al. (2019) and Mataev et al. (2019) employ extra regularization to boost performance of the deep image prior. Chen et al. (2020b); Ho et al. (2020) leverage neural architecture search to obtain a better deep image prior network for improved performance. Asim et al. (2019) employ deep image prior on image patches, which improves its reconstruction ability. Zukerman et al. (2020) improve the deep image prior by using a backprojection loss function. These approaches improve results, but still require an oracle to determine when to stop the optimization. In this paper, we boost the performance of the deep image prior by controlling its spectral bias, and achieve an automatic stopping with a new criterion.

An intuition provided by Ulyanov et al. (2018, 2020) for the workings of the deep image prior is that their network follows an encoder-decoder architecture, which imposes strong priors about natural images. Heckel and Soltanolkotabi (2020) further attribute the effects of the deep image prior to the special architecture with convolutions using fixed interpolating filters. Alternatively, Cheng et al. (2019) explain the deep image prior from a Bayesian perspective by showing that the model behaves like a stationary Gaussian process at initialization. These works have focused on studying the workings of deep image prior, mostly from the view of the network architecture design. In this paper, we provide a complementary perspective. We show that the spectral bias leads the networks to capture deep image priors during optimization, beyond the choice of the architecture. We do so by introducing a metric, the Frequency Band Correspondence, which offers a spectral measurement of the deep image prior, revealing the low-frequency natural image signals are learned faster and better than high-frequency noise signals.

A downside of the original deep image prior (Ulyanov et al. 2018, 2020) is the requirement of an oracle to determine when to stop the optimization as its performance degrades after reaching a peak over the iterations of optimization. Heckel and Hand (2019) tackle this problem with an underparameterized network, at the expense of reduced performance. Cheng et al. (2019) avoid the need for early stopping with a Bayesian approach, at the expense of slower convergence. In this paper, we prevent the performance degradation over iterations with Lipschitz-controlled spectral bias and enable stopping the optimization automatically at an appropriate moment with a new criterion.

A few recent works (Rahaman et al. 2019; Xu et al. 2020; Chakrabarty and Maji 2019) have paid attention to the spectral bias as well. Rahaman et al. (2019) and Xu et al. (2020) analyze the spectral bias for classification problems with supervised learning, not for generative models with a single image. Chakrabarty and Maji (2019) exposed the deep image prior has a spectral bias by adding noise at different frequencies to the image and analyzing the optimization trajectories from different noisy versions of the input. However, they do not measure and control the bias. In this work, we propose a frequency band correspondence to measure the spectral bias of the deep image prior. We further control the bias to address the performance degradation problem and the performance-convergence trade-off problem.

3 Measuring Spectral Bias

The literature attributes the ability of an untrained network to obtain restored results from degraded target images to a particular architecture, like an encoder-decoder, which imposes strong priors about natural images. In this work, we show that the spectral bias leads the networks to capture deep image priors during optimization, beyond the choice of the architecture. We do so by introducing a metric, the Frequency Band Correspondence, which offers a spectral measurement of the deep image prior, revealing the low-frequency natural image signals are learned faster and better than high-frequency noise signals, and pinpoint why inverse images can be restored, when the network optimization is stopped at the right time.

3.1 Frequency-Band Correspondence Metric

The proposed Frequency-Band Correspondence metric examines the input-output correspondence in the frequency domain across several frequency bands. For this metric, let $\{\theta^{(1)}, \ldots, \theta^{(T)}\}$ denote the trajectory of T steps of gradient descent in the parameter space and let $\{f_{\theta^{(1)}}, \ldots, f_{\theta^{(T)}}\}$ denote the corresponding trajectory in the output space. We propose to analyze the Fourier spectrum of the output images $f_{\theta^{(t)}, t=1,\ldots,T}$ to show the convergence dynamics of different frequency components of the target image. The Fourier spectrum of the output image $f_{\theta^{(t)}}$ is obtained by the Fourier transform $\mathcal{F}$, denoted as $\mathcal{F}\{f_{\theta^{(t)}}\}$ for step t. We similarly compute the Fourier transform for the target image y_0, denoted as $\mathcal{F}\{y_0\}$. We then compute an element-wise correspondence between both transforms as:

$$H_{\theta^{(t)}} = \frac{\mathcal{F}\{f_{\theta^{(t)}}\}}{\mathcal{F}\{y_0\}}. \tag{1}$$

Intuitively, $H_{\theta^{(t)}}$ denotes to what extent any deep image prior at step t corresponds with image y_0 in the frequency domain; the closer the values are to 1, the higher the correspondence. As we are interested in the spectral bias of the deep image prior, we divide the correspondence map into N subgroups corresponding to N non-overlapping frequency bands. Since the correspondence map is symmetrical around the center, we group it according to the distance between its elements and its center uniformly, as illustrated in Fig. 1. To transform the 2D map to the 1D one, we compute the mean correspondence for each band, denoted as $\bar{H}^{(n)}_{\theta^{(t)}}$, with $n=1, \ldots, N$. The value of $\bar{H}^{(n)}_{\theta^{(t)}}$ indicates the convergence dynamics of different frequency components of a target image.

3.2 Spectral Measurement of Deep Image Prior

We use this metric, denoted as FBC (Frequency-Band Correspondence), to measure how well the network output of the deep image prior corresponds to the target image as a function of N frequency bands. Since the FBC metric is computed with the Fourier transform, our spectral measurement in this

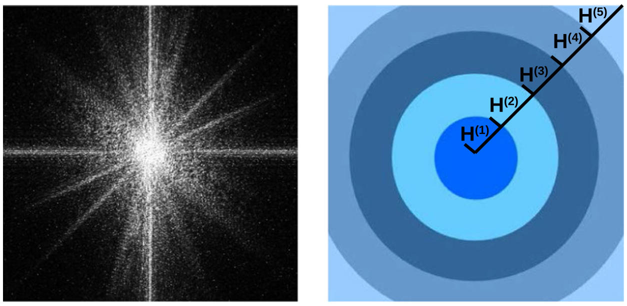


Fig. 1 Frequency-band correspondence metric. The left image shows an example of correspondence map H, which is computed according to Eq. (1). We divide the correspondence map into N subgroups corresponding to N non-overlapping frequency bands. Since the correspondence map is symmetrical around the center, we group it according to the distance between its elements and its center uniformly, as illustrated by the right image when $N = 5$. Different colors represent different subgroups. We compute the mean correspondence for each band to transform the 2D map to the 1D one

section denotes the frequency domain analysis. The Fourier transform $\mathcal{F}$ in Eq. (1) is implemented by means of the 2D Fast Fourier Transform, where only the magnitude is used to compute the Fourier spectrum of the images. We use $N{=}5$ where frequency bands are divided into the lowest frequency, low frequency, medium frequency, high frequency and the highest frequency. We perform empirical studies on three inverse imaging problems, including image denoising, JPEG image deblocking, and image inpainting with the ‘peppers’, ‘F16’ and ‘Lena’, images from Dabov et al. (2007). For image denoising, the image is degraded by adding Gaussian noise with two noise levels, including $\sigma{=}15$ and $\sigma{=}25$, following Zhang et al. (2017). Following Dong et al. (2015a), we evaluate JPEG image deblocking on the gray-scale images, which are compressed with the PIL encoder into two quality levels, including $quality{=}10$ and $quality{=}20$. For image inpainting, the image is degraded by using a central region mask, and we consider two hole-to-image area ratios, including $ratio{=}0.1$ and $ratio{=}0.25$, following Pathak et al. (2016). Following Ulyanov et al. (2018, 2020), the network input is given as uniform noise between 0 and 0.1 with a depth of 32 by default.

First, we investigate whether the network of the original deep image prior exhibits any form of spectral bias in its optimization. We take the *Encoder-Decoder* architecture of Ulyanov et al. (2018, 2020) and show its Frequency Band Correspondences for five frequency bands in Fig. 3. The plot highlights, across inverse imaging problems, degradation levels and degraded images, low frequencies are learned quickly and with high correspondence to the target image, while high frequencies are learned slower and with lower correspondence. We conclude that the network of the deep image prior during optimization has a spectral bias towards low frequencies, and this bias helps to obtain a meaningful performance. The peak PSNR (Peak Signal-to-Noise Ratio) performance of the deep image prior occurs when the lowest frequencies are matched nearly perfect, while the highest frequencies are less used, as marked by the green vertical lines. However, once the higher frequencies obtain a higher correspondence, the performance starts to drop.

Next, we show that such a spectral bias is not specific to the *Encoder-Decoder* architecture. We take two other architectures as examples, as shown in Fig. 2. We remove the Encoder from the *Encoder-Decoder* architecture of Ulyanov et al. (2018, 2020) to obtain the *Decoder*. We additionally remove the upsampling layers from the *Decoder* to get the *ConvNet*. Figure 4a and b show that both *Decoder* and *ConvNet* learn low-frequency components of the target image faster than learning the high-frequency components, reaffirming the spectral bias. We also observe that *ConvNet* learns high-frequency components faster than *Decoder* by remov-

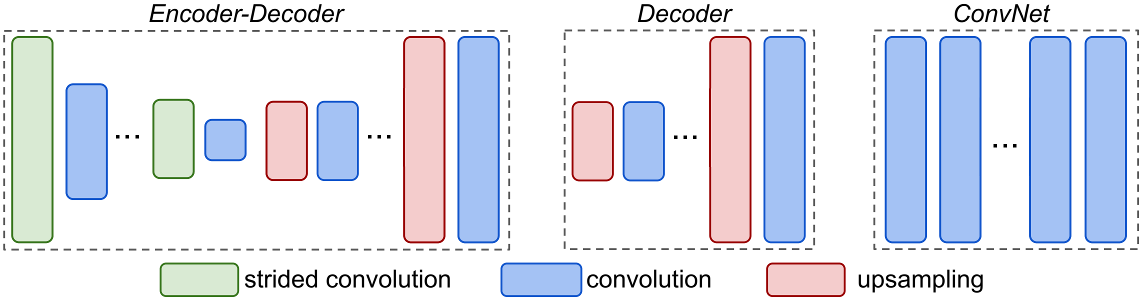
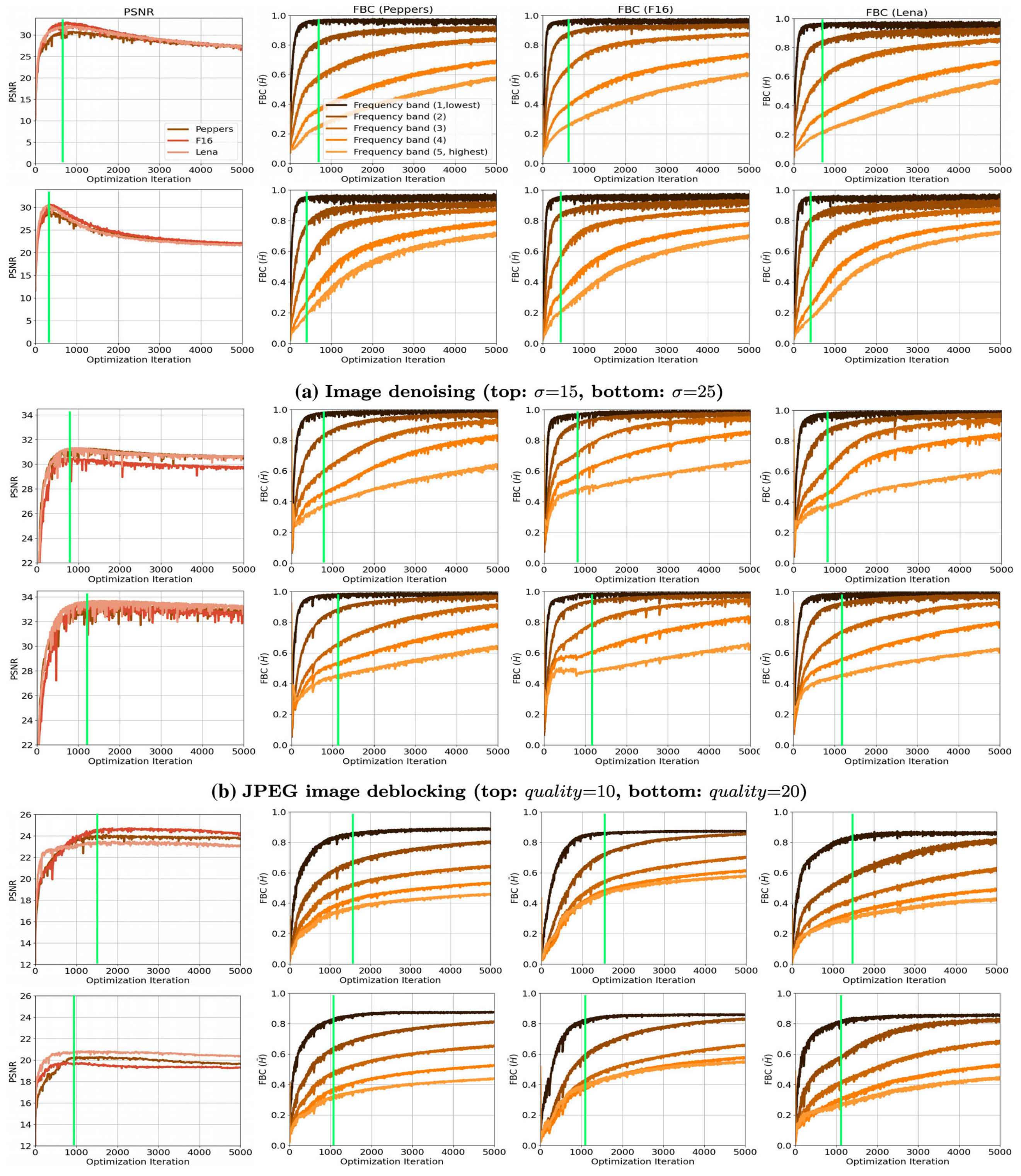


Fig. 2 Network architectures used in the experiments of Sect. 3. The *Encoder-Decoder* is the same as the one used in Ulyanov et al. (2020). Specifically, the encoder contains five convolution blocks. Each block contains two convolution layers with the kernel size of 3 × 3 and the channel number of 128. The stride of the first convolution layer is set to 2 to achieve the downsampling. The decoder contains five bilinear upsampling layers, where each upsampling layer is followed by a convolution layer with the kernel size of 3 × 3 and the channel number of 128. Each convolution layer is followed by a batch normalization layer and a leaky ReLU layer with a negative slope of 0.01. The *Decoder* is obtained by removing the encoder from the *Encoder-Decoder*. Removing the upsampling layers from the *Decoder* finally leads to the *ConvNet*

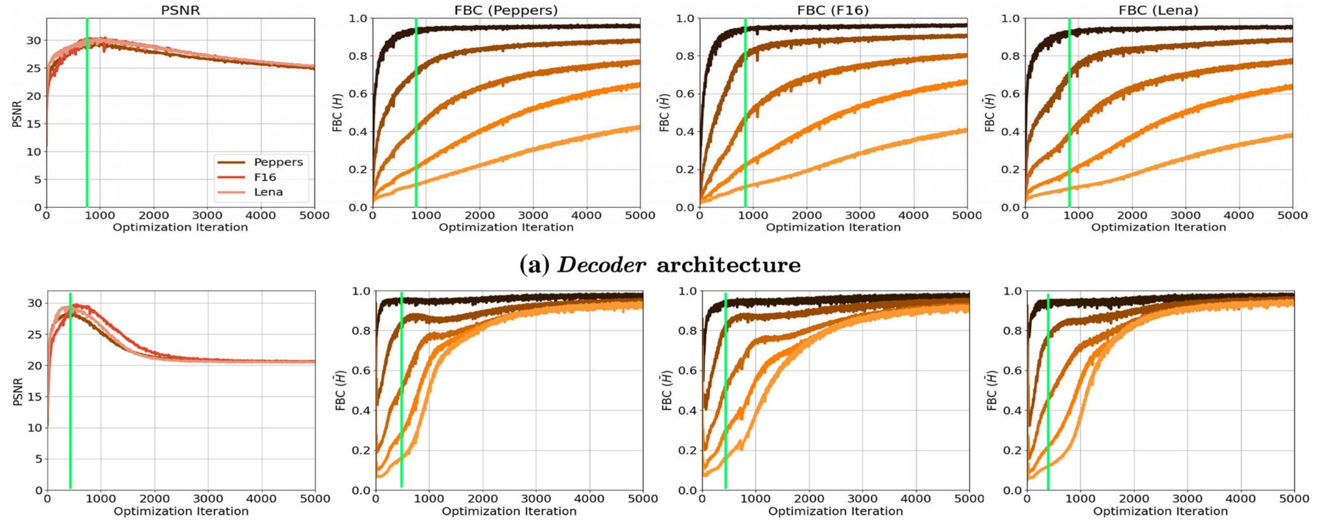


(a) Image denoising (top: σ=15, bottom: σ=25)

(b) JPEG image deblocking (top: $quality$=10, bottom: $quality$=20)

(c) Image inpainting (top: $ratio$=0.1, bottom: $ratio$=0.25)

Fig. 3 Spectral measurement of the deep image prior on image denoising, JPEG image deblocking and image inpainting. The network of the deep image prior (Ulyanov et al. 2020) exhibits a spectral bias during optimization across inverse imaging problems, degradation levels and degraded images, where lower frequencies are learned faster and better than high-frequencies. The degraded images can be restored well when optimizations are stopped at the right time, as marked by the green vertical lines (Color figure online)



(a) ***Decoder*** **architecture**

(b) ***ConvNet*** **architecture**

Fig. 4 **Spectral measurement of the deep image prior** with different architectures on image denoising. The spectral bias is not specific to the *Encoder-Decoder* architecture of Ulyanov et al. (2020). Alternative architectures, such as a *Decoder* and a *ConvNet*, also exhibit a bias towards specific image frequencies during optimization. Also, the *ConvNet* learns higher frequencies faster than the *Decoder* by removing the upsampling layers, but at the expense of reduced peak performance

ing the upsampling layers, but at the expense of reduced peak performance. Having established the architecture is not critical for the deep image prior, we use from now on the *Decoder* as the default network architecture to benefit from a good trade-off between performance and run-time.

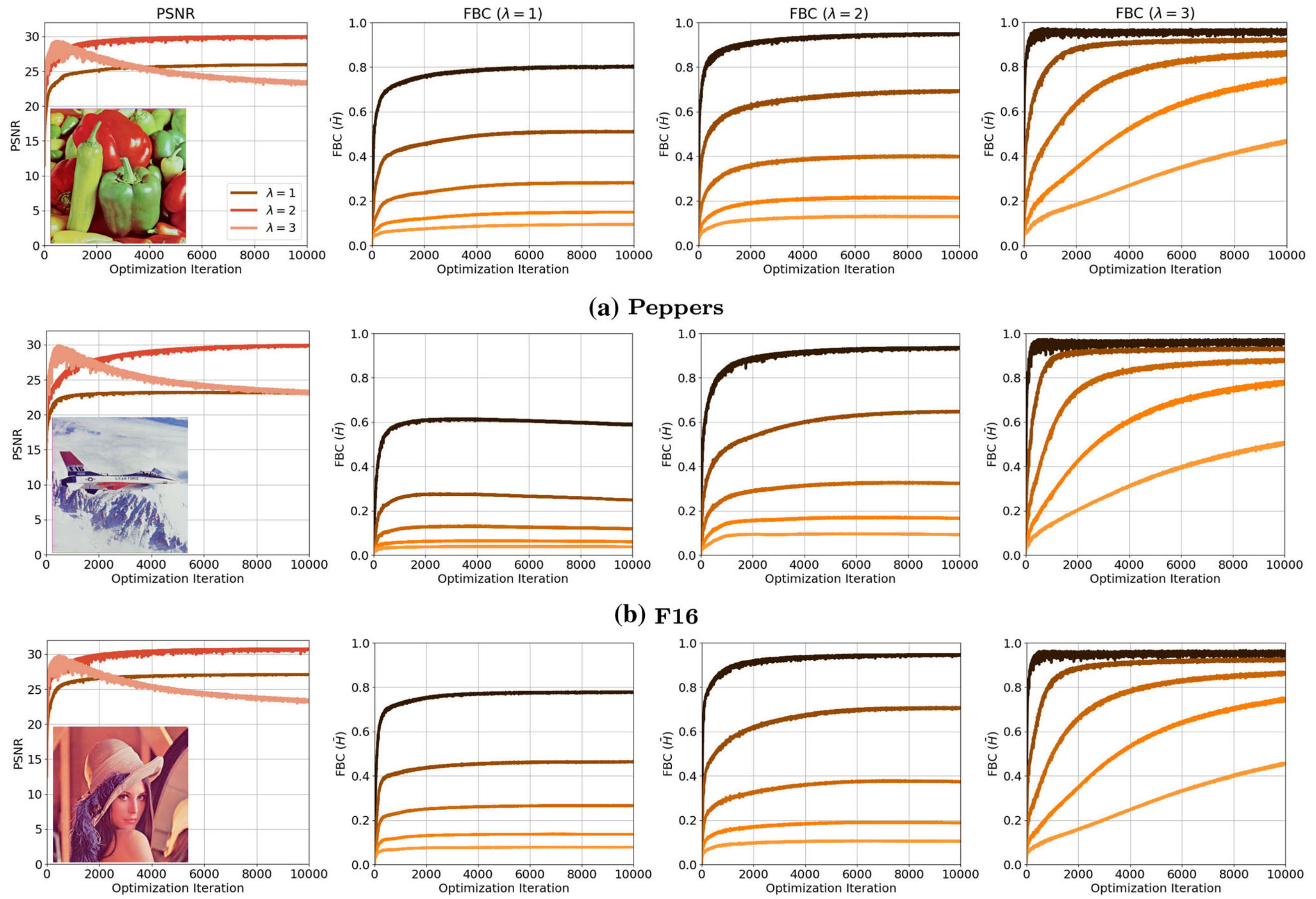
Our study provides a clear implication: untrained solutions for inverse imaging problems work by a latent ability to learn low frequencies faster than learning high frequencies. As natural images are well approximated by low-frequency components, degraded images can be restored well when optimizations are stopped at the right time. The network is optimized to fit the degraded image, in which higher frequencies consist of both structured high-frequency image details and random high-frequency noise. The structured high-frequency image details, that have self-similarity across the image, are fitted better and faster. However, once the random high-frequency noise is fitted over a certain level, which could affect the structured high-frequency image details, the output quality degrades. This behavior explains why the performance in the deep image prior degrades when training longer. Hence, a key enabler for improving the deep image prior is to control the spectral bias by restricting the fitting of random high-frequency noise in the output. Our study also finds that the upsampling layer is beneficial for obtaining good peak performance, but may introduce too much spectral bias towards the low frequencies, slowing down the learning of desired high frequencies. Hence, it's a feasible way to balance peak performance and convergence by controlling the spectral bias in upsampling.

4 Controlling Spectral Bias

We exploit the measured spectral bias to avoid the degradation of performance over iterations and to balance peak performance and convergence. We do so by controlling spectral biases in the two core layer types of inverse imaging networks: the convolution layer and the upsampling layer. We present a Lipschitz-controlled approach for the convolution and a Gaussian-controlled approach for the upsampling layer. The approaches are general in their setup, making them applicable to any network form and scale. Besides these two methods for controlling spectral bias, we further introduce a simple stopping criterion to avoid superfluous computation.

4.1 Lipschitz-Controlled Spectral Bias

From the point of view of the frequency domain, the Fourier spectrum of the network indicates its ability to learn higher frequencies. Lower frequencies are learned first, while higher frequencies are learned later in the optimization process. This implies that the ability of the network to learn higher frequencies is gradually enhanced by optimizing the learnable layers. Improving the Fourier spectrum of the network is only achievable through adjusting the spectrum of the learnable layers. Based on this observation, we aim to upper bound the Fourier coefficients of the convolutional layers, for the sake of constraining the Fourier spectrum of the network. We are able to impose an upper bound on the Fourier coefficients of a convolution layer by enforcing Lipschitz continuity, accord-



(a) Peppers

(b) F16

(c) Lena

Fig. 5 **Lipschitz-controlled spectral bias** for image denoising. Setting the right Lipschitz constant (λ=2) avoids performance decay while maintaining a high PSNR. Different constants result in different levels of spectral bias. A high constant (λ=3) still incorporates a lot of high-frequency noise signals, while a low constant (λ=1) fails to incorporate the important low frequency image signals. With the right balance (λ=2), we maintain the low frequencies while avoiding the high-frequency noise signals

ing to Katznelson (2004). Specifically, if a convolution layer f is Lipschitz continuous, there exists a constant L for any inputs x, y satisfying $\| f(x) - f(y)\| \leq L\|x - y\|$. The minimum over all such values satisfying this condition is called the Lipschitz constant of f, denoted by C. Then the Fourier coefficients of f, *i.e.*, $|\hat{f}(\boldsymbol{k})|$, is bounded by,

$$|\hat{f}(\boldsymbol{k})| \leq \frac{C}{|\boldsymbol{k}|^2}. \tag{2}$$

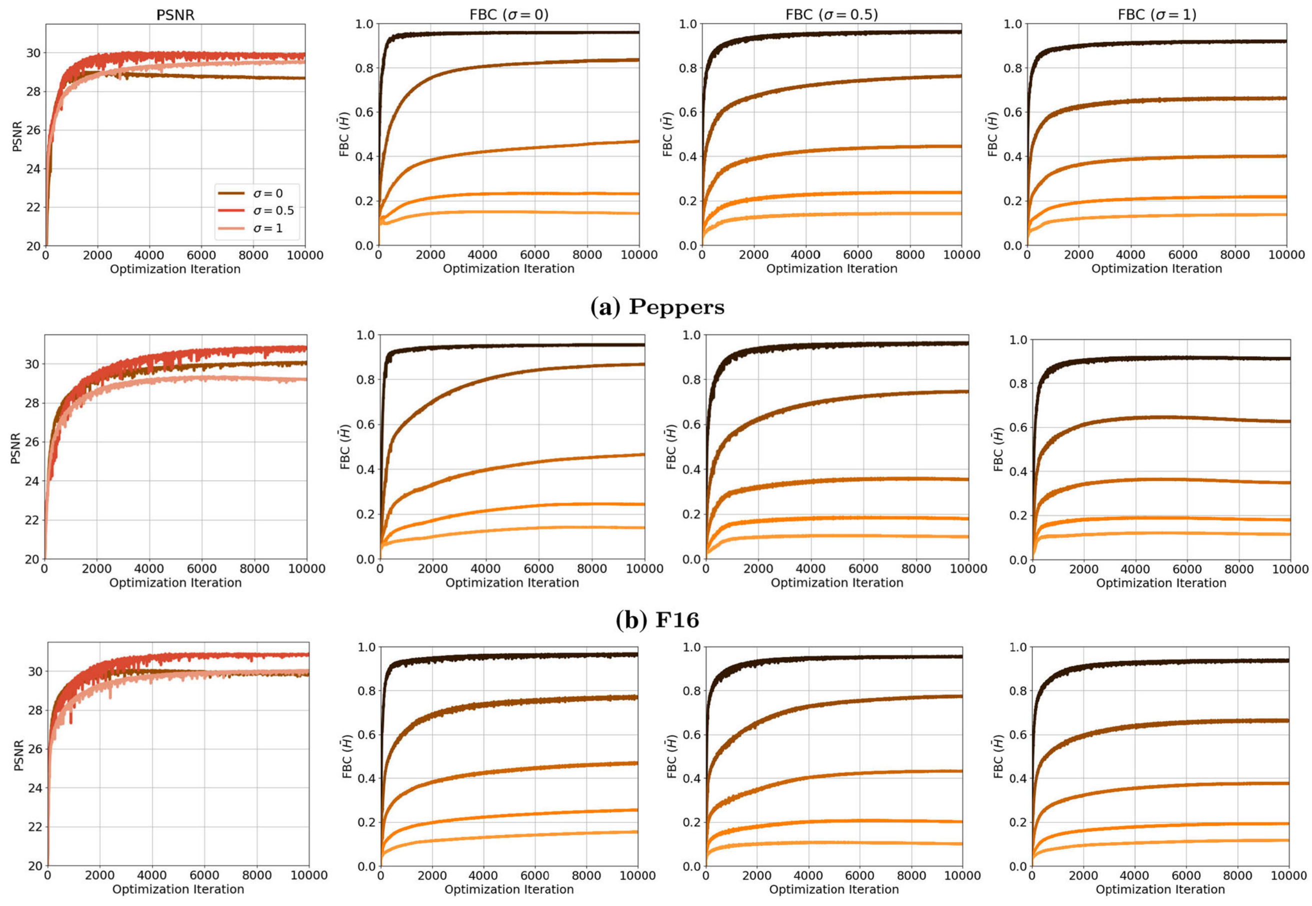
Further, the Lipschitz constant of a convolution layer is bounded by the spectral norm of its parameters. Then we obtain,

$$|\hat{f}(\boldsymbol{k})| \leq \frac{C}{|\boldsymbol{k}|^2} \leq \frac{\|\boldsymbol{w}\|_{sn}}{|\boldsymbol{k}|^2}, \tag{3}$$

where $\boldsymbol{w}$ is the weight of a convolution layer f, and $\|\cdot\|_{sn}$ denotes the spectral norm, which can be approximated relatively quickly using a few iterations of the power method (Miyato et al. 2018). The power law $|\boldsymbol{k}|^{-2}$ indicates that the spectral decay is stronger towards higher frequencies, which means that learning higher frequencies requires a higher spectral norm. Thus, we are able to manipulate the ability of a convolution layer in learning higher frequencies by upper bounding its spectral norm to a specific value λ with $\frac{\boldsymbol{w}}{\max(1, \|\boldsymbol{w}\|_{sn}/\lambda)}$. Where we leave the weight matrix $\boldsymbol{w}$ untouched if its spectral norm is lower than λ. Otherwise, we normalize $\boldsymbol{w}$ by $\|\boldsymbol{w}\|_{sn}/\lambda$.

To accelerate and stabilize the optimization, batch normalization (Ioffe and Szegedy 2015) is often used after convolution layers. However, we find it is not compatible with our Lipschitz constraining as its output is invariant to the channel weight vector norm $\|\boldsymbol{w}\|_p$, *i.e.*,

$$BN(\boldsymbol{w}\boldsymbol{x}/\|\boldsymbol{w}\|_p) = BN(\boldsymbol{w}\boldsymbol{x}), \tag{4}$$



(a) Peppers

(b) F16

(c) Lena

Fig. 6 **Gaussian-controlled spectral bias** for image denoising. Varying the Gaussian kernel by σ controls convergence and performance. Too small values (σ=0) results in worse performance, while too big values (σ=1) introduce too much smoothing, slowing down the convergence. With a suitable value (σ=0.5), our upsampling introduces an appropriate spectral bias, leading to fast convergence and good denoising performance

where $\boldsymbol{x}$ denotes the channel input. We therefore propose a Lipschitz normalization by exploring the idea of combining Lipschitz constraining with a special version of batch normalization: mean-only batch normalization. We only subtract out the minibatch means, without dividing by the minibatch standard deviations. The Lipschitz normalization is defined as:

$$\mathrm{LN}(\boldsymbol{w}, \boldsymbol{x}) = \frac{\boldsymbol{wx}}{\max(1, \|\boldsymbol{w}\|_{sn}/\lambda)} - \mu + b, \tag{5}$$

where μ denotes the channel mean of the pre-activation $\boldsymbol{wx}$ and b is a scalar bias term. The Lipschitz normalization layer is inserted between a convolutional layer and a ReLU activation. With this normalization, the Lipschitz constant of a convolution layer is bounded by the hyperparameter λ. As a result, we can manipulate the ability of the network in learning high frequencies by tuning λ, leading to a controlled spectral bias of the deep image prior.

4.2 Gaussian-Controlled Spectral Bias

Upsampling is an important operation in network architectures for inverse imaging problems, as it produces high-resolution outputs from low-resolution inputs. Well-known approaches such as the bilinear and nearest neighbor upsampling have a constant smoothing effect (Chakrabarty and Maji 2019; Heckel and Soltanolkotabi 2020). Different tasks, however, might operate best under different levels of smoothing. Too strong a smoothing introduces too much spectral bias towards lower frequencies. This slows down the learning of the desired higher frequencies, delaying convergence of optimization (as shown in Fig. 4). Therefore, we propose an upsampling method which allows controlling the amount of smoothing and is capable of balancing performance and convergence.

We first decompose the upsampler into an expansion and a filtering step. Let x_i be the i-th channel of input x. For

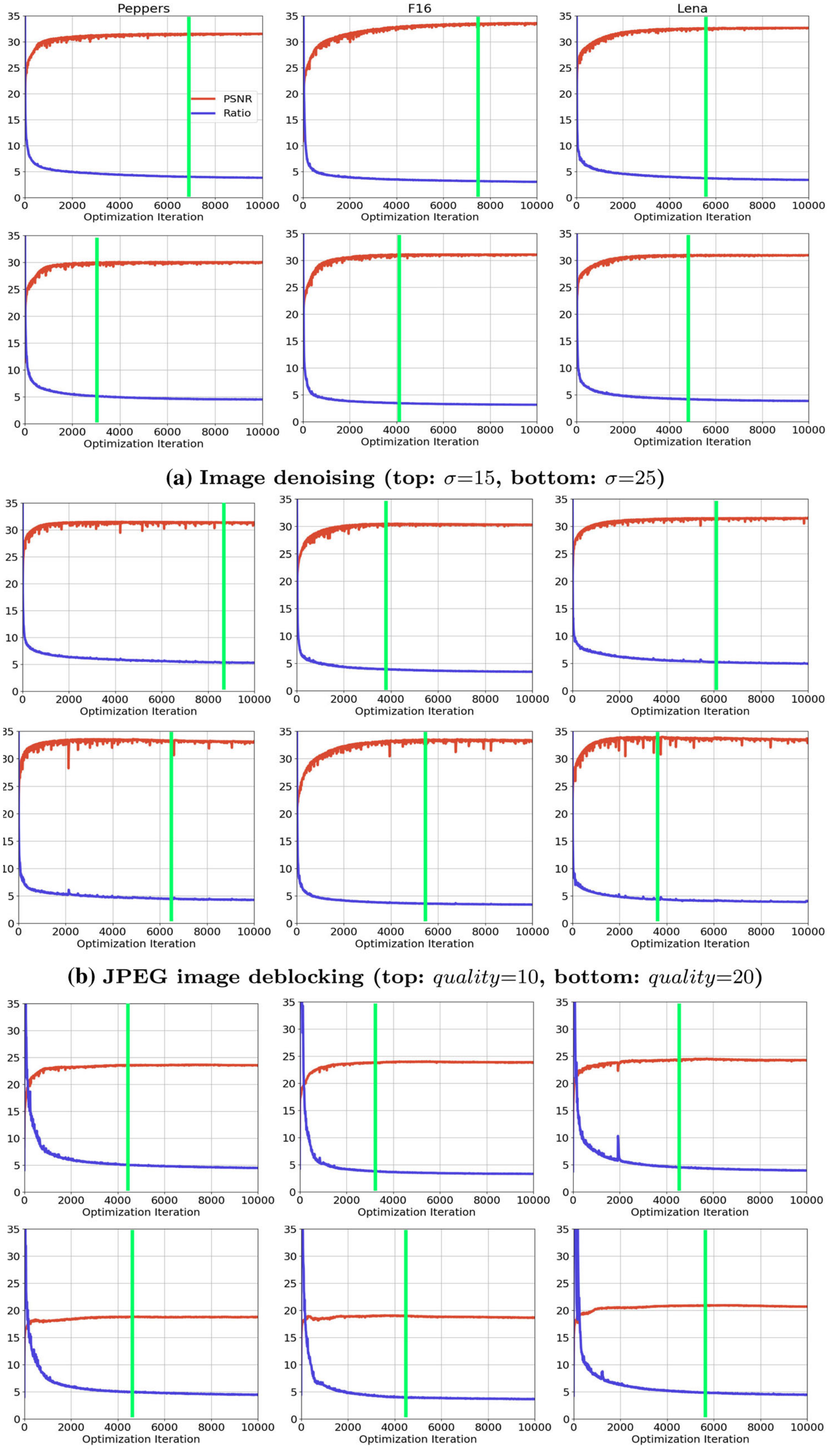


Fig. 7 Automatic stopping criterion evaluated on image denoising, JPEG image deblocking and image inpainting. The vertical green line shows the selected iteration by the proposed stopping criterion. Across inverse imaging problems, degradation levels and degraded images, we observe the optimization can be stopped earlier, with a minimal performance loss compared to a fixed stop at 10,000 iterations

(a) Image denoising (top: σ=15, bottom: σ=25)

(b) JPEG image deblocking (top: *quality*=10, bottom: *quality*=20)

(c) Image inpainting (top: *ratio*=0.1, bottom: *ratio*=0.25)

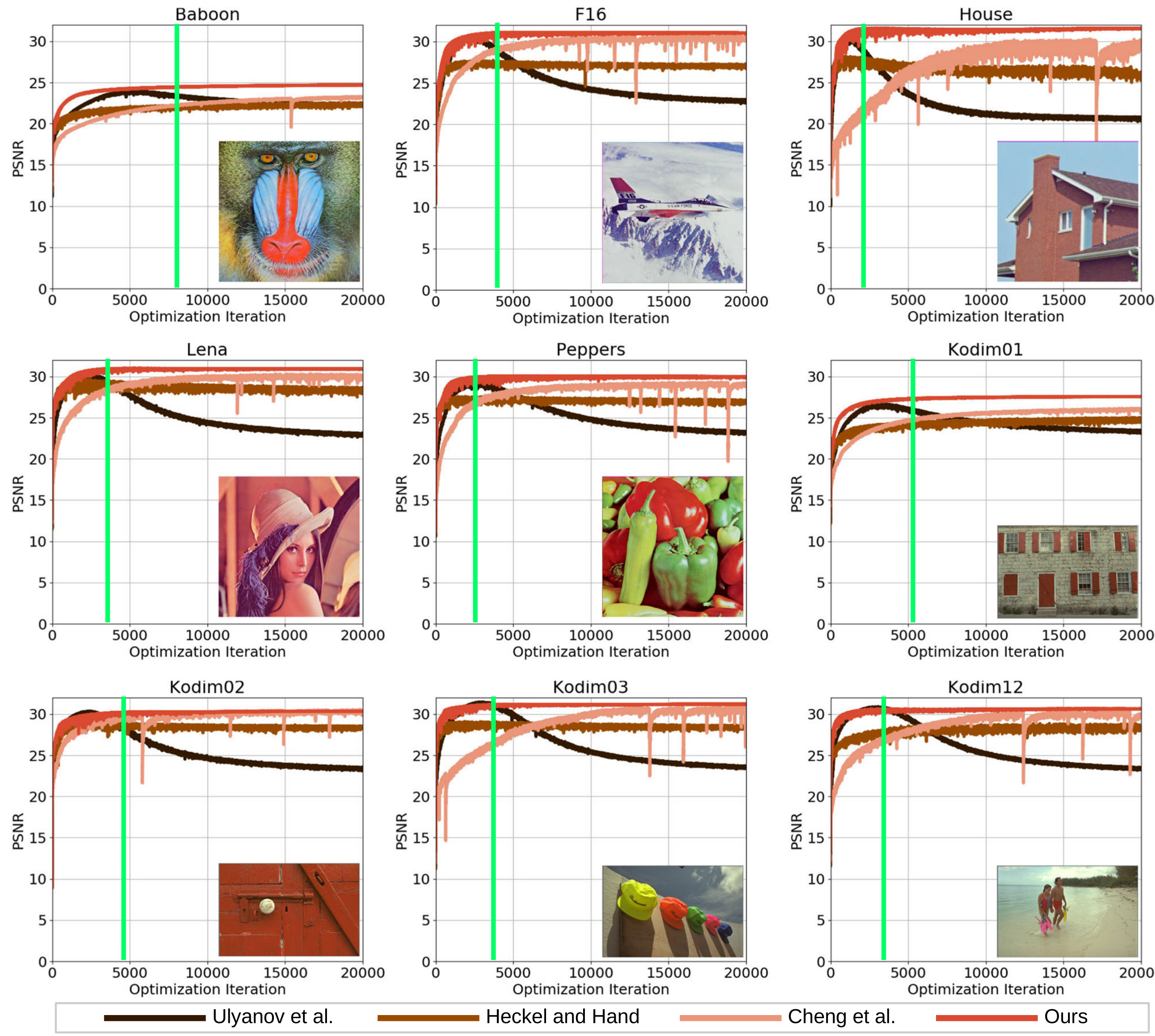


Fig. 8 **Image denoising.** PSNR scores of various methods over multiple iterations for removing additive Gaussian white noise with σ=25. Compared to (Ulyanov et al. 2020), our method doesn't suffer from performance degradation, and we can stop the optimization automatically at an appropriate moment for each image (marked by the green vertical lines), leading to good PSNR scores. Compared to Heckel and Hand (2019) and Cheng et al. (2019), we either achieve a faster convergence or obtain a higher PSNR score

expansion, x_i is padded with a "bed of nails" scheme, *i.e.*, inserting $s-1$ zeros between the pixels of x_i along its rows and columns. Such a "bed of nails" expansion creates a high-frequency replica of the original signal. To smooth out the noisy high-frequencies, we perform filtering by convolving the upsampled signal with an interpolating filter. We use a Gaussian filter sampled by $\mathcal{N}(0, \sigma^2)$. Hence, we define our Gaussian upsampling by:

$$\text{Up}(x_i) = \uparrow_s (x_i) * G_\sigma, \tag{6}$$

where $\uparrow_s (x_i)$ denotes expanding x_i with factor s, $*$ is the convolution operation, G_σ denotes the Gaussian filter. In the frequency domain, we obtain the Fourier spectrum of our upsampling by,

$$\mathcal{F}(\text{Up}(x_i)) = \mathcal{F}(\uparrow_s (x_i)) \odot \mathcal{F}(G_\sigma), \tag{7}$$

where $\mathcal{F}$ denotes the Fourier transform, $\odot$ is the Hadamard product and $\mathcal{F}(G_\sigma)[k]{=}1/e^{2\pi^2\sigma^2k^2}$. We manipulate the Fourier spectrum of our upsampling by choosing different σ, allowing us to control the spectral bias in the upsampling.

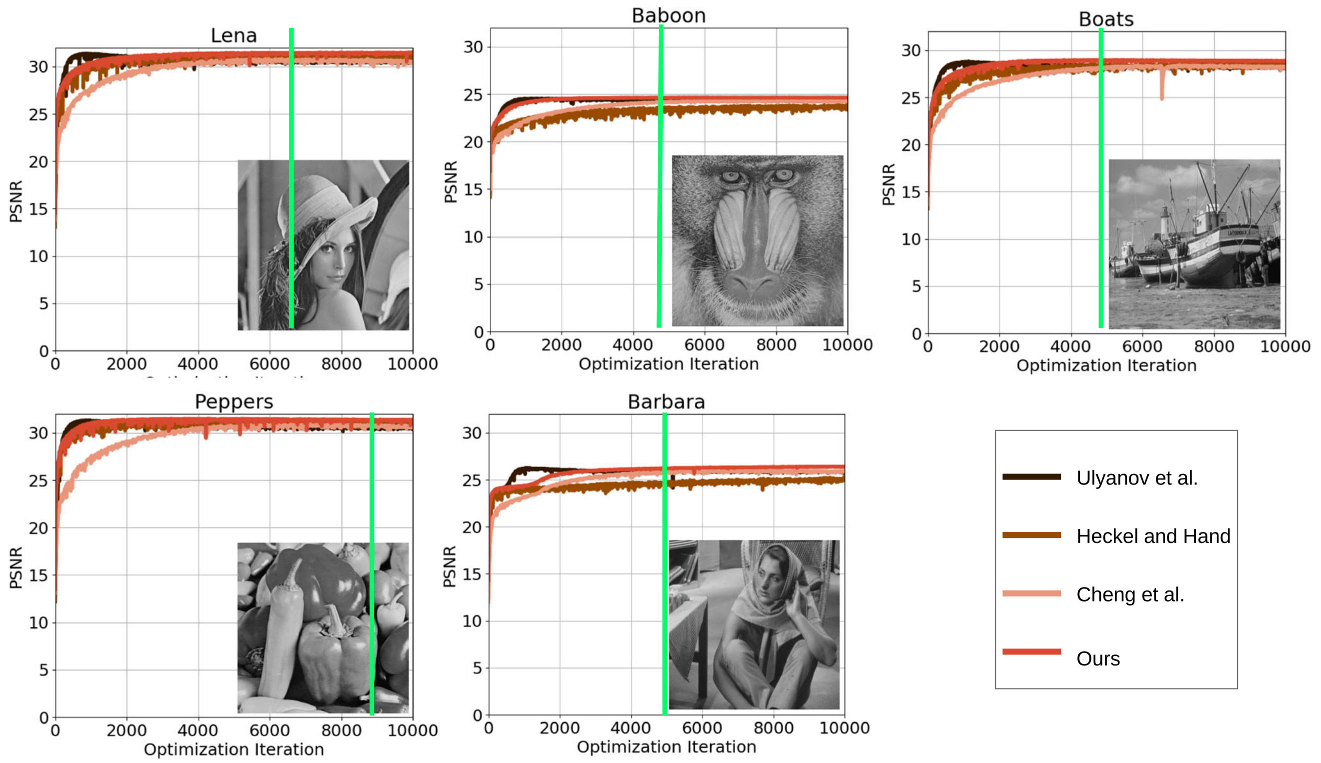


Fig. 9 **JPEG image deblocking.** PSNR scores of various methods when reducing artifacts of a compressed JPEG image with $quality$=10. We again observe that the performance of the deep image prior (Ulyanov et al. 2020) degrades. Cheng et al. (2019) and Heckel and Hand (2019) do not suffer from degradation, at the expense of either reduced performance or slow convergence. Our method achieves a good trade-off between PSNR score and convergence (marked by the green vertical lines)

4.3 Automatic Stopping Criterion

With the ability to control the spectral bias, we can fix the number of iterations for network optimization without fear of performance degradation. As different tasks have different levels of convergence, however, using a fixed number of iterations still leads to redundant optimization. To improve efficiency, we introduce a simple criterion to automatically perform early stopping.

It is well known that an image looks blurry when there is a high amount of low frequencies in its Fourier spectrum. We exploit this property by computing the blurriness and sharpness for an output image and use their ratio as the metric to stop the optimization. In case of a spectral bias, low frequencies will be learned first, while high-frequencies will be learned later. Our Lipschitz normalization limits the ability of the network in learning high frequencies to an upper bound. Hence, when this upper bound is reached, the ratio of blurriness to sharpness of the output image will converge as well. To that end, we design the following measure:

$$\begin{aligned} r(f_\theta) &= \mathcal{B}(f_\theta)/\mathcal{S}(f_\theta), \\ \Delta r({f_\theta}^t) &= \left| \frac{1}{n}\sum_{i=1}^{n} r\left({f_\theta}^{(t-i)}\right) - \frac{1}{n}\sum_{i=1}^{n} r\left({f_\theta}^{(t-n-i)}\right) \right|, \end{aligned} \quad (8)$$

where f_θ denotes the output image and ${f_\theta}^{(t)}$ denotes an instance in iteration t. $\mathcal{B}(f_\theta)$ denotes the blurriness of the output image y computed using Crete et al. (2007). $\mathcal{S}(f_\theta)$ denotes the sharpness of the output image y computed using Bahrami and Kot (2014). $r(f_\theta)$ denotes the ratio of blurriness to sharpness of the output image f_θ. Then, $\frac{1}{n}\sum_{i=1}^{n} r\left({f_\theta}^{(t-i)}\right)$ computes the mean ratio of output images from iteration $(t-1)$ to $(t-n)$, and $\frac{1}{n}\sum_{i=1}^{n} r\left({f_\theta}^{(t-n-i)}\right)$ computes the mean ratio of output images from iteration $(t-n-1)$ to $(t-2n)$. If their absolute difference is smaller than a constant value ϵ, the optimization is stopped.

Compared to the ratio r itself, the ratio difference Δr between optimization iterations is independent of the images. Since the deep image prior no longer suffers from performance degradation with the controlled spectral bias, the ratio r barely changes when the performance is stable. Thus, we can set the ratio difference threshold ϵ to a small value, like

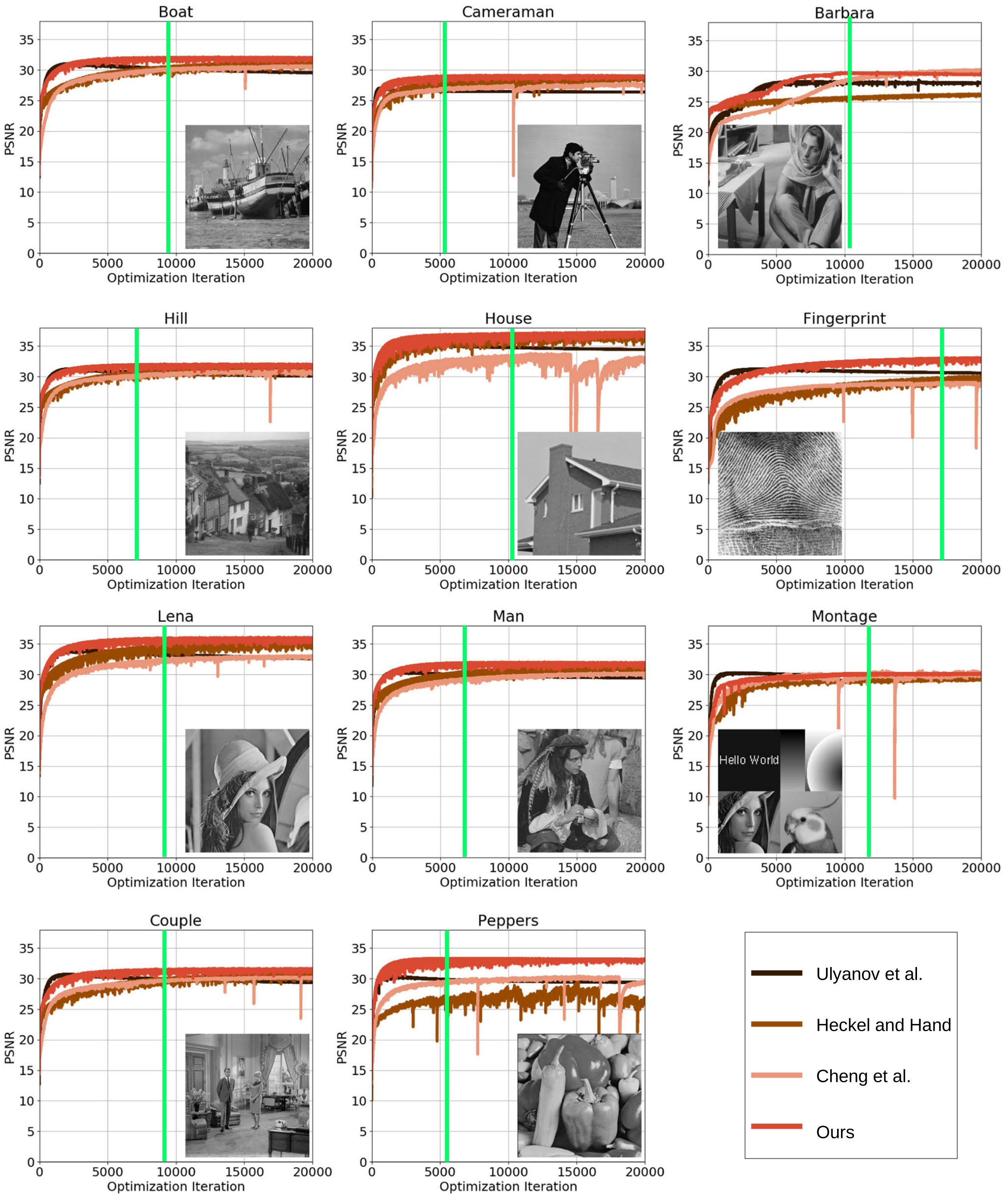


Fig. 10 **Image inpainting.** PSNR scores of various methods for pixel inpainting. We again observe the degradation of performance over iterations for the deep image prior (Ulyanov et al. 2020). Cheng et al. (2019) and Heckel and Hand (2019) do not suffer from the degradation problem, at the expense of either reduced performance or slow convergence. Our method achieves a good trade-off between PSNR score and convergence (marked by the green vertical lines) (Color figure online)

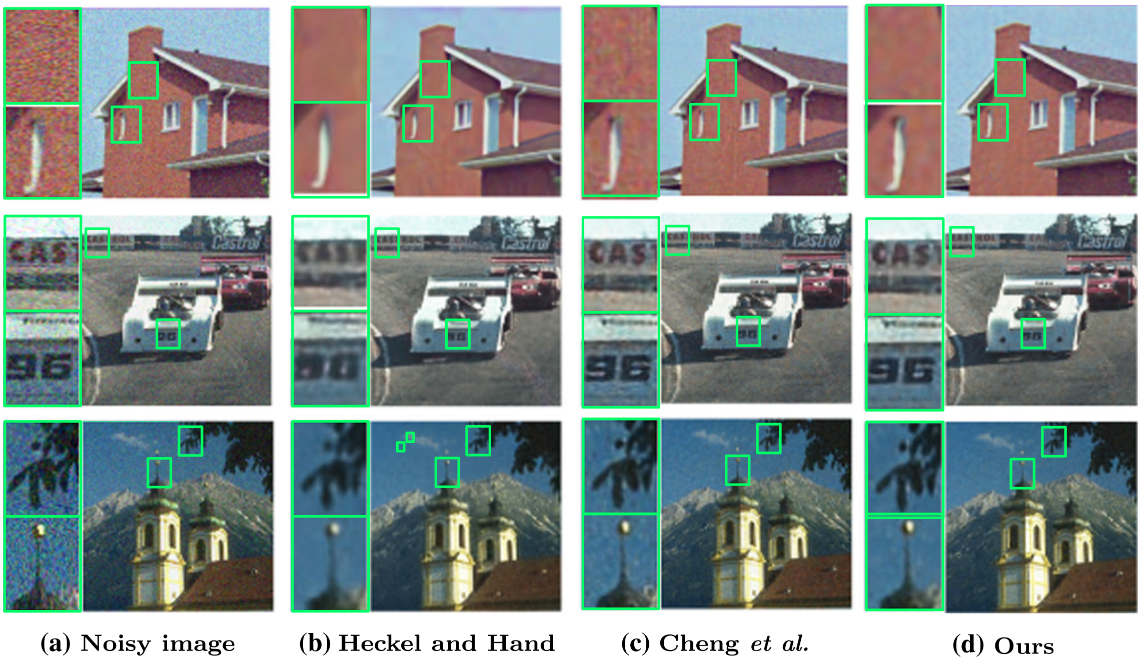


Fig. 11 **Image denoising**. The goal is to remove the additive Gaussian white noise with σ=25. From the top regions masked by the green rectangles, we observe the method of Cheng et al. (2019) still overfits some high-frequency noise, while our method does not. From the bottom regions masked by the green rectangles, we observe the method of Heckel and Hand (2019) has difficulty preserving high-frequency edges, while our method performs better

0.01. As the main benefit of the auto-stopping is to avoid redundant computation, it does not directly affect the inverse imaging performance. Note that the stopping criterion fails for the original deep image prior (Ulyanov et al. 2018, 2020) because the high-frequency components of its output image keeps increasing until the degraded target image is fully fitted.

4.4 Performance Analysis

We empirically analyze the deep image prior with the Lipschitz-controlled spectral bias, the Gaussian-controlled spectral bias and the automatic stopping criterion.

Lipschitz-controlled spectral bias. Following the work of Ulyanov et al. (2018, 2020), we use bilinear upsampling in this experiment. In Eq. (5), λ is the only parameter which controls the ability of the network in learning high frequencies. Finding the best λ for each image is still an open question. Here we just empirically study three settings, *i.e.*, λ=1,λ=2, and λ=3. The spectral norm $\|\boldsymbol{w}\|_{sn}$ is estimated with the power iteration method (Miyato et al. 2018). The results are shown in Fig. 5. Setting a suitable constraint (*e.g.*, λ=2) results in a PSNR curve without performance decay. The FBC graphs show this is because setting a low Lipschitz constant amplifies the spectral bias. High frequencies are hardly incorporated at all, while low frequencies still obtain a high correspondence to the target image. Using a too high constraint (*e.g.*, λ=3) results in a similar performance peak and decay as the original deep image prior. When using a too low constraint (*e.g.*, λ=1), we not only suppress high frequencies, but also the low frequencies, which generally corresponds to the structure of the image, hampering the performance. We conclude, utilizing Lipschitz normalization with a suitable value of λ addresses the problem of performance degradation.

Table 1 **Image denoising** on CBSD68 for varying σ

	15	25	50
Ulyanov et al. (2020)[†*]	30.58	27.84	24.59
Heckel and Hand (2019)[†]	28.66	26.60	24.06
Cheng et al. (2019)[†]	30.77	28.08	24.71
Ours	**30.80**	**28.15**	**24.83**
CBM3D (Dabov et al. 2007)[††]	33.52	30.71	27.38
CDnCNN (Zhang et al. 2017)[††]	33.89	31.23	27.92
FFDNet (Zhang et al. 2018)	33.87	31.21	27.96

Supervised approaches and CBM3D prevail, but our unsupervised method obtains better PSNR than the deep image prior and its variants across three noise levels

[†] Results based on author-provided code

[*] Results obtained with oracle stopping

[††] Results provided by Zhang et al. (2018)

Gaussian-controlled spectral bias. Next, we study the effect of the Gaussian-controlled spectral bias to balance performance and convergence. We replace the bilinear upsampling with our Gaussian upsampling and use λ=2 to maintain the effect of the Lipschitz-controlled spectral bias on avoiding performance degradation. We consider Gaussian upsampling with three settings in Eq. (6), σ=0, σ=0.5, σ=1 where the kernel size is fixed to 5 × 5. We show the effect of different settings on the denoising performance and amount of spectral bias in Fig. 6. The smaller the value for σ, the faster the convergence is reached. However, a too small value *e.g.*, σ=0 results in worse performance, because the upsampling reduces to the "bed of nails" expansion. A value of σ=1 introduces too much smoothing, slowing down the convergence. With a suitable value, *e.g.*, σ=0.5, our upsampling introduces an appropriate spectral bias, leading to fast convergence and good denoising performance. Furthermore, compared to the widely used upsampling, like bilinear upsampling (refer to its performance in Fig. 5), our upsampling achieves a better trade-off between performance and convergence. We conclude our upsampling allows to control the spectral bias, enabling us to improve the performance of deep image prior for inverse imaging problems like image denoising.

Table 2 **JPEG image deblocking** on LIVE1 for varying quality levels

	10	20	30
Ulyanov et al. (2020)[†*]	27.52	29.75	31.08
Heckel and Hand (2019)[†]	26.63	27.65	28.99
Cheng et al. (2019)[†]	27.59	29.81	31.12
Ours	**27.70**	**29.86**	**31.14**
AR-CNN (Dong et al. 2015a)[††]	28.96	31.29	32.67
TNRD (Chen and Pock 2016)[††]	29.15	31.46	32.84
DnCNN (Zhang et al. 2017)	29.19	31.59	32.98

Supervised approached prevail, but compared to the unsupervised deep image prior and its variants, our method obtains better performance in terms of PSNR

[†] Results based on author-provided code

[*] Results obtained with oracle stopping

[††] Results provided by Zhang et al. (2017)

Stopping criterion. Finally, we analyze the effect of the proposed stopping criterion on image denoising, JPEG image deblocking and image inpainting. For each problem, we evaluate on different degradation levels, as specified before in Sect. 3.2. We use n=100 and ϵ=0.01 throughout the experiment. We set the fixed stopping iteration to 10,000. We show

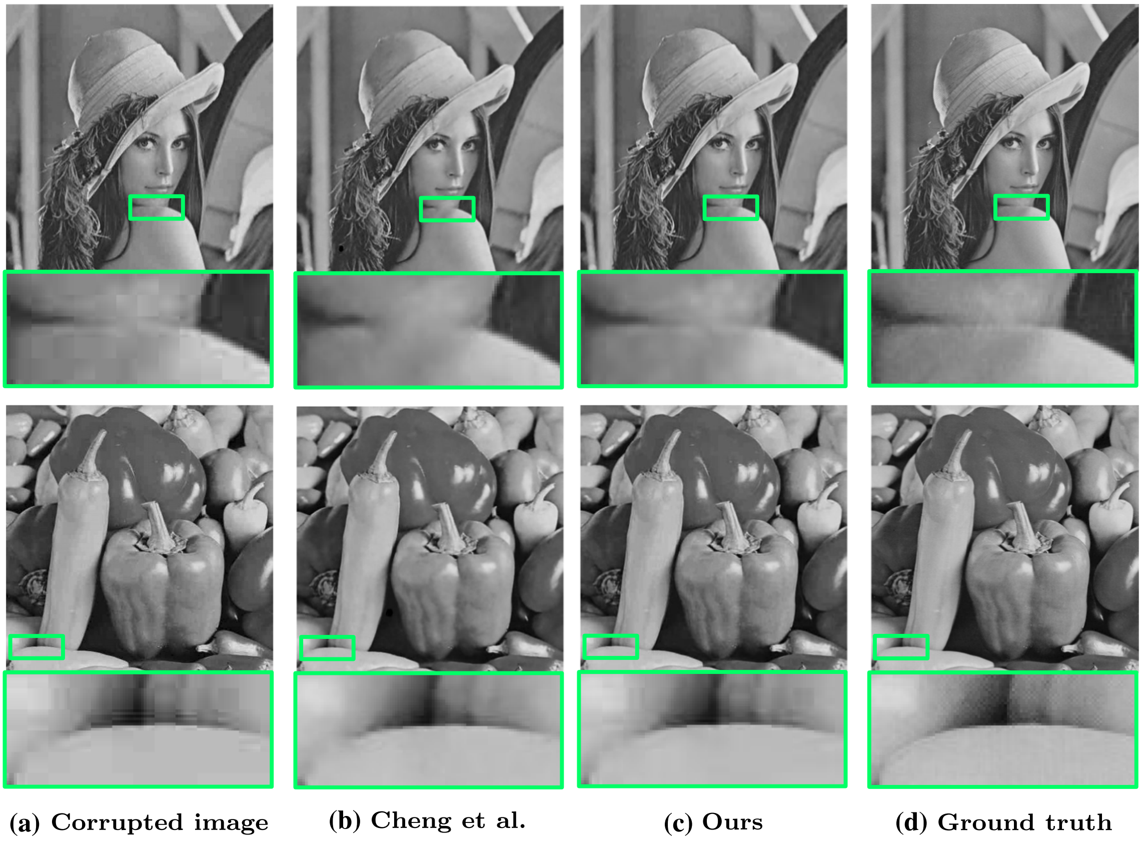


Fig. 12 **JPEG image deblocking**. The goal is to reduce the artifacts of the compressed JPEG image with $quality$=20. From the regions masked by the green rectangles, we observe our method performs well, especially when reducing the artifacts and recovering high-frequency image details

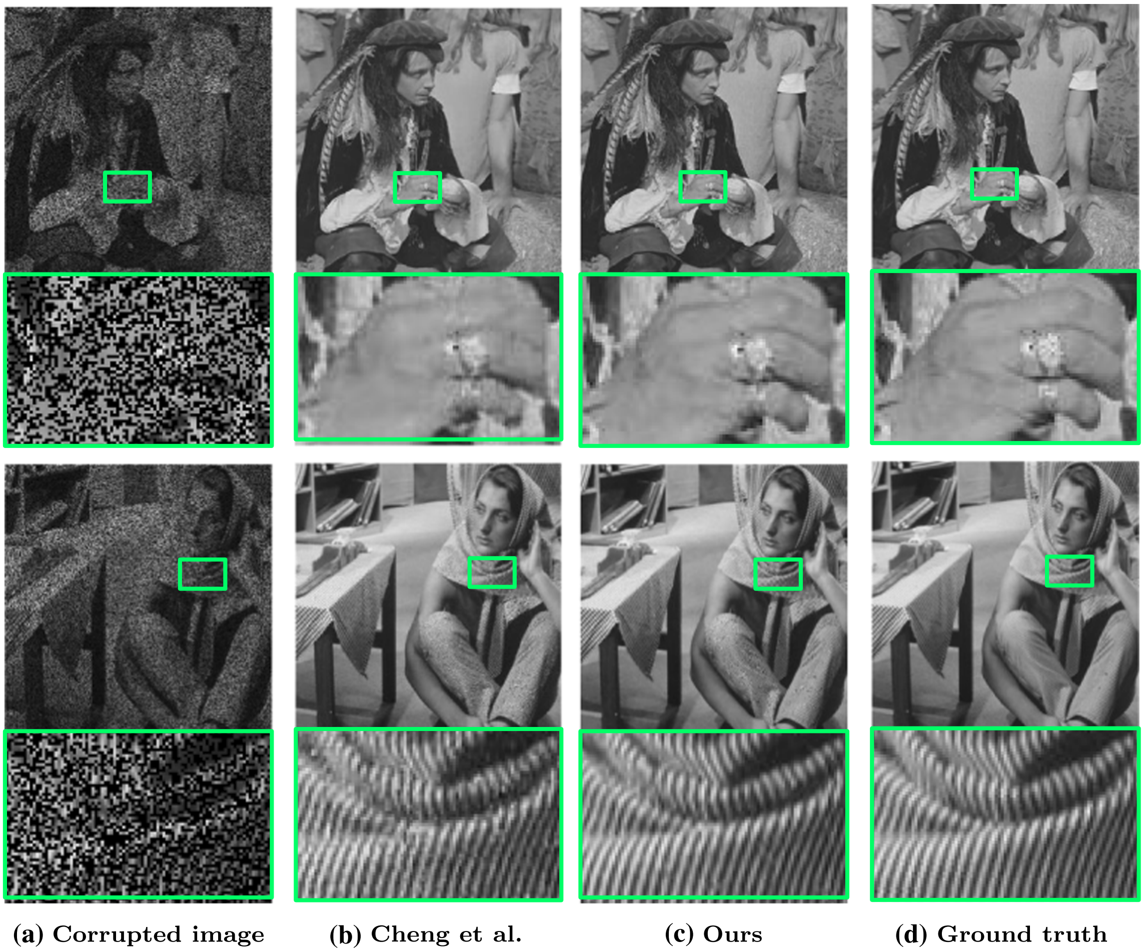


Fig. 13 **Image inpainting**. The goal is to reconstruct the 50% missing pixels resulting from a binary Bernoulli mask. From the regions masked by the green rectangles, we observe our method performs well, especially when recovering high-frequency details

the dynamics of the Peak Signal-to-Noise score and ratio values in Fig. 7. We observe the stopping criterion is effective, it reduces the number of required iterations considerably with only a minimal loss in performance, across inverse imaging problems, degradation levels, and degraded images. For the worst performing "F16" image for denoising with $\sigma=25$, the PSNR drops from 31.04 to 30.98 when reducing the iterations from 10,000 to 3,896. We also found that the performance in terms of PSNR changes less than 0.1 when the ratio difference threshold ϵ ranges from 0.001 to 0.1. A bigger threshold means the optimization stopped earlier.

Table 3 **Image inpainting** on CBSD68 for varying ratio

	0.1	0.25	0.5
Ulyanov et al. (2020) †*	22.78	19.42	17.26
Heckel and Hand (2019)†	21.52	18.67	16.81
Cheng et al. (2019)†	22.83	19.49	17.28
Ours	**22.87**	**19.58**	**17.36**

In terms of PSNR, our method outperforms the deep image prior and its variants on region-based inpainting, across three hole-to-image area ratios

†Results based on author-provided code

*Results obtained with oracle stopping

5 Applications

With the gained ability to control the spectral bias in the deep image prior, we consider four inverse imaging applications and one image enhancement application for comparative evaluation: image denoising, JPEG image deblocking, image inpainting, image super-resolution and image detail enhancement. On all tasks, we compare to the deep image priors of Ulyanov et al. (2018, 2020), Heckel and Hand (2019) and Cheng et al. (2019). For reference, we also report the results obtained by classical methods like (Dabov et al. 2007), and supervised-learning based methods like (Zhang et al. 2017).

We report our results with the *Decoder*, introduced in Sect. 3.2, as our network architecture. Lipschitz normaliza-

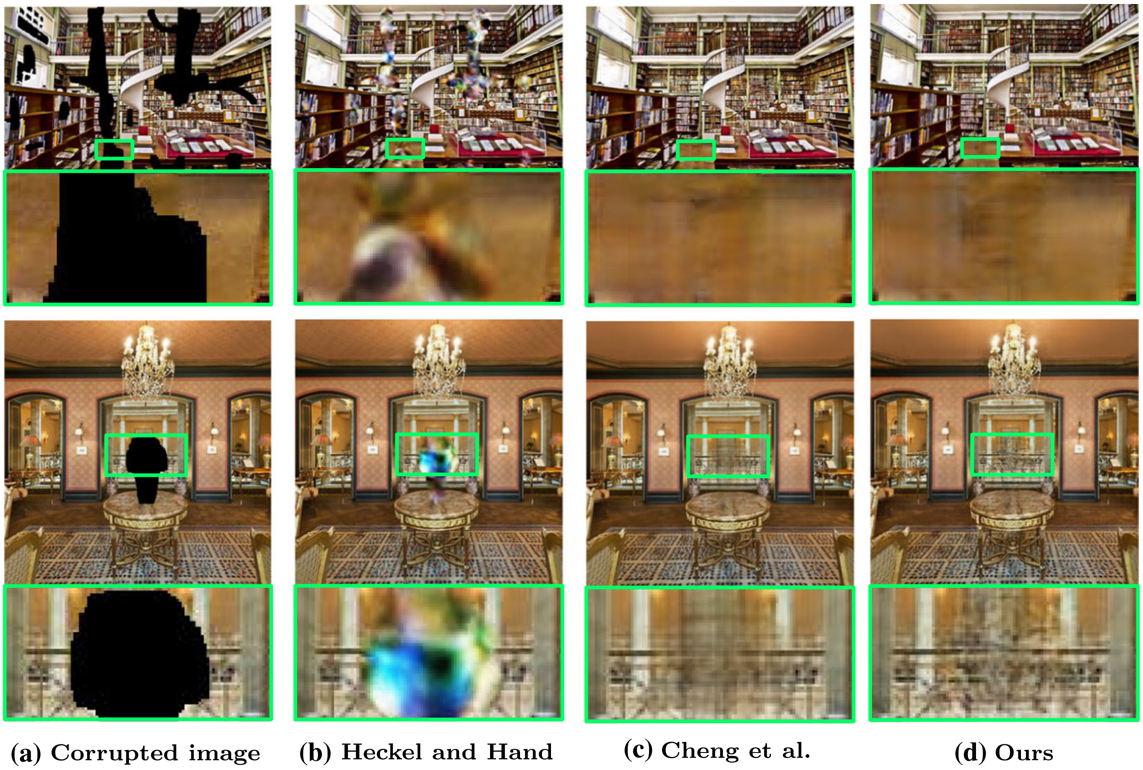


Fig. 14 **Image inpainting**. The goal is to reconstruct the missing pixels resulting from a binary region mask. From the regions masked by the green rectangles, we observe our method performs better than Heckel and Hand (2019) and as good as Cheng et al. (2019)

tion with λ=2 and Gaussian upsampling with σ=0.5 are combined into the *Decoder* to achieve a controllable deep image prior. Network parameters are initialized with He initialization (He et al. 2015). Our approach works with popular optimizers such as standard gradient descent and Adam (Kingma and Ba 2015). Following Ulyanov et al. (2018, 2020), we use Adam with a mini-batch of 1 to optimize our networks. We set β_1 to 0.9, β_2 to 0.999 and the initial learning rate to 0.001. The network input is a uniform noise between 0 and 0.1 with a depth of 32 by default. Our code will be released.

5.1 Image Denoising

For the denoising comparison we use two datasets, *i.e.*, the standard dataset by Dabov et al. (2007) consisting of 9 RGB images, and CBSD68 by Roth and Black (2009) consisting of 68 RGB images. Each noisy image is generated by adding an additive Gaussian white noise with three noise levels, including σ=15, σ=25 and σ=50. The goal is to distill the original image without Gaussian noise. Results on the dataset of Dabov et al. (2007) are shown in Fig. 8, where PSNR scores of various methods are shown over multiple iterations. The performance of the deep image prior (Ulyanov et al. 2018, 2020) gradually degrades after reaching a peak. For each image, the peak is reached at a different number of iterations, so simply using a fixed number of iterations will be suboptimal for most images.

Our method provides two advantages: (1) The performance does not decay over iterations with controlled spectral bias; (2) The optimization can be automatically stopped at an appropriate moment using the proposed stopping criterion, leading to good PSNR scores for all images (marked by the green vertical lines). Heckel and Hand (2019) achieve fast convergence without performance degradation, but at the expense of reduced performance. Cheng et al. (2019) obtain comparable PSNR scores, but they require 2 to 4 times as many iterations to converge.

So far, we have shown the performance of various methods per image over a varying number of optimization iterations. Next, we compare their overall PSNR performance on the 68 images in CBSD68, as shown in Table 1. While our unsupervised method is outperformed by supervised-learning alternatives (Zhang et al. 2017, 2018) and CBM3D (Dabov et al. 2007), it does better than the deep image prior (Ulyanov et al. 2018, 2020), and its variants (Heckel and Hand 2019; Cheng et al. 2019) across three noise levels. We also provide qualitative results for denoising in Fig. 11, where we observe our method preserves the high-frequency edges without overfitting to high-frequency noise.

Table 4 **Super-resolution** on Set14. The PSNR scores are reported for a stopping iteration of 2000 for the scaling of 4, and 4000 for the scaling of 8, following Ulyanov et al. (2018, 2020)

4× **resolution**	Baboon	Barbara	Bridge	Coastguard	Comic	Face	Flowers	Foreman	Lenna	Man	Monarch	Pepper	Ppt3	Zebra	*Average*
Ulyanov et al. (2020)*	22.29	25.53	24.38	25.81	22.18	31.02	26.14	31.66	30.83	26.09	29.98	32.08	24.38	25.71	27.00
Heckel and Hand (2019)†	20.54	21.51	20.97	23.52	18.86	28.15	20.88	24.44	24.07	21.18	21.21	23.89	17.28	18.59	21.79
Cheng et al. (2019)†	21.51	24.84	23.74	25.02	21.94	30.11	25.41	30.57	28.62	25.37	28.41	30.25	23.69	24.48	26.07
Ours	**22.81**	**25.74**	**25.02**	**25.86**	**22.31**	**32.09**	**26.74**	**32.77**	**31.29**	**26.42**	**30.77**	**32.62**	**24.73**	**25.87**	**27.50**
Bicubic††	22.44	25.15	24.47	25.53	21.59	31.34	25.33	29.45	29.84	25.70	27.45	30.63	21.78	24.01	26.05
TV prior††	22.34	24.78	24.46	25.78	21.95	31.34	25.91	30.63	29.76	25.94	28.46	31.32	22.75	24.52	26.42
LapSRN (Lai et al. 2018)††	22.83	25.69	25.36	26.21	22.90	32.62	27.54	33.59	31.98	27.27	31.62	33.88	25.36	26.98	28.13
8× **resolution**															
Ulyanov et al. (2020)*	21.38	23.94	22.20	**24.21**	19.86	29.52	22.86	27.87	**27.93**	23.57	**24.86**	**29.18**	20.12	**20.62**	24.15
Heckel and Hand (2019)†	20.07	19.86	19.67	22.30	18.01	26.53	19.57	22.76	22.06	19.54	19.71	21.44	15.64	17.20	20.31
Cheng et al. (2019)†	19.81	23.69	22.19	19.22	19.72	28.88	22.81	27.34	19.69	23.36	24.43	28.72	26.14	19.89	20.67
Ours	**21.57**	**24.48**	**22.64**	24.18	**19.71**	**29.94**	**22.92**	**27.95**	27.67	**23.86**	24.46	28.91	**23.28**	19.93	**24.17**
Bicubic††	21.28	23.44	22.24	23.65	19.25	28.79	22.06	25.37	26.27	23.06	23.18	26.55	18.62	19.59	23.09
TV prior††	21.30	23.72	22.30	23.82	19.50	28.84	22.50	26.07	26.74	23.53	23.71	27.56	19.34	19.89	23.48
LapSRN (Lai et al. 2018)††	21.51	24.21	22.77	24.10	20.06	29.85	23.31	28.13	28.22	24.20	24.97	29.22	20.13	20.28	24.35

On most images we achieve better performance than existing methods, and we obtain the highest PSNR on average for 4× and 8× super-resolution

†Results based on author-provided code

*Results obtained with oracle stopping

††Results provided by Ulyanov et al. (2020)

Table 5 **Super-resolution** on set5. The PSNR scores are reported for a stopping iteration of 2000 for the scaling of 4, and 4000 for the scaling of 8, following Ulyanov et al. (2020)

4× **resolution**	Baby	Bird	Butterfly	Head	Woman	*Average*
Ulyanov et al. (2020) *	31.49	31.80	26.23	31.04	**28.93**	29.89
Heckel and Hand (2019)[†]	24.57	24.66	18.46	27.64	22.44	23.55
Cheng et al. (2019)[†]	27.35	28.37	24.21	27.45	25.48	26.57
Ours	**32.76**	**32.71**	**26.47**	**31.79**	28.54	**30.45**
Bicubic [††]	31.78	30.20	22.13	31.34	26.75	28.44
TV prior [††]	31.21	30.43	24.38	31.34	26.93	28.85
LapSRN (Lai et al. 2018) [††]	33.55	33.76	27.28	32.62	30.72	31.58
8× **resolution**						
Ulyanov et al. (2020) *	28.28	**27.09**	20.02	29.55	24.50	25.88
Heckel and Hand (2019)[†]	21.95	22.97	16.18	26.56	20.46	21.62
Cheng et al. (2019)[†]	27.18	26.64	19.64	24.76	23.81	24.41
Ours	**28.46**	26.79	**20.32**	**30.07**	**24.76**	**26.08**
Bicubic [††]	27.28	25.28	17.74	28.82	22.74	24.37
TV prior [††]	27.93	25.82	18.40	28.87	23.36	24.87
LapSRN (Lai et al. 2018) [††]	28.88	27.10	19.97	29.76	24.79	26.10

On most images we achieve better performance than existing methods, and we perform best on average for both 4× and 8× super-resolution

[†] Results based on author-provided code

*Results obtained with oracle stopping

[††] Results provided by Ulyanov et al. (2020)

5.2 JPEG Image Deblocking

JPEG image deblocking is the process of reducing the compression artifacts in JPEG images. We evaluate on the Classic5 dataset by Foi et al. (2006) and the LIVE1 dataset by Sheikh et al. (2006). Classic5 consists of 5 gray-scale images, and LIVE1 consists of 29 color images. Following Dong et al. (2015a), the color images are transformed to gray-scale using the YCbCr color model by keeping the Y component only. Then, the gray-scale images are compressed with the PIL encoder into three qualities, 10, 20, and 30. Fig. 9 provides a quantitative comparison on Classic5 for $quality$=10. Akin to the denoising comparison, we again observe the degradation of performance over iterations for the deep image prior (Ulyanov et al. 2018, 2020). Cheng et al. (2019) and Heckel and Hand (2019) do not suffer from the degradation problem, at the expense of either reduced performance or slow convergence. With the controlled spectral bias and automatic stopping criterion, we achieve a good trade-off between PSNR score and convergence (marked by the green vertical lines).

We also provide quantitative results for LIVE1 in Table 2. Naturally, the learning-based methods (Dong et al. 2015a; Chen and Pock 2016; Zhang et al. 2017) perform best. Across three quality levels, our unsupervised method performs better than the deep image prior (Ulyanov et al. 2018, 2020) and its two variants (Heckel and Hand 2019; Cheng et al. 2019). We also provide qualitative examples in Fig. 12, which shows that our method better reduces the artifacts and recovers high-frequency image details.

5.3 Image Inpainting

In image inpainting, we are given an image with missing pixels resulting from a binary mask. The goal is to reconstruct the missing data. We evaluate on the standard dataset by Heide et al. (2015), consisting of 11 grayscale images, and the CBSD68 dataset by (Roth and Black 2009) consisting of 68 RGB images. Following Ulyanov et al. (2018, 2020); Cheng et al. (2019), we consider inpainting with masks that are randomly sampled according to a binary Bernoulli distribution on the standard dataset. Each mask is sampled to drop 50% of the pixels at random. For CBSD68, we consider inpainting with central region masks and we evaluate on three hole-to-image area ratios, $ratio$=0.1, $ratio$=0.25 and $ratio$=0.5, following Pathak et al. (2016). Figure 10 provides a quantitative comparison on the standard dataset. We also provide quantitative results for CBSD68 in Table 3. Our observations are the same as for the denoising and deblocking comparison. We provide qualitative examples for pixel inpainting in Fig. 13 and region inpainting in Fig. 14, which shows our ability to recover high-frequency details.

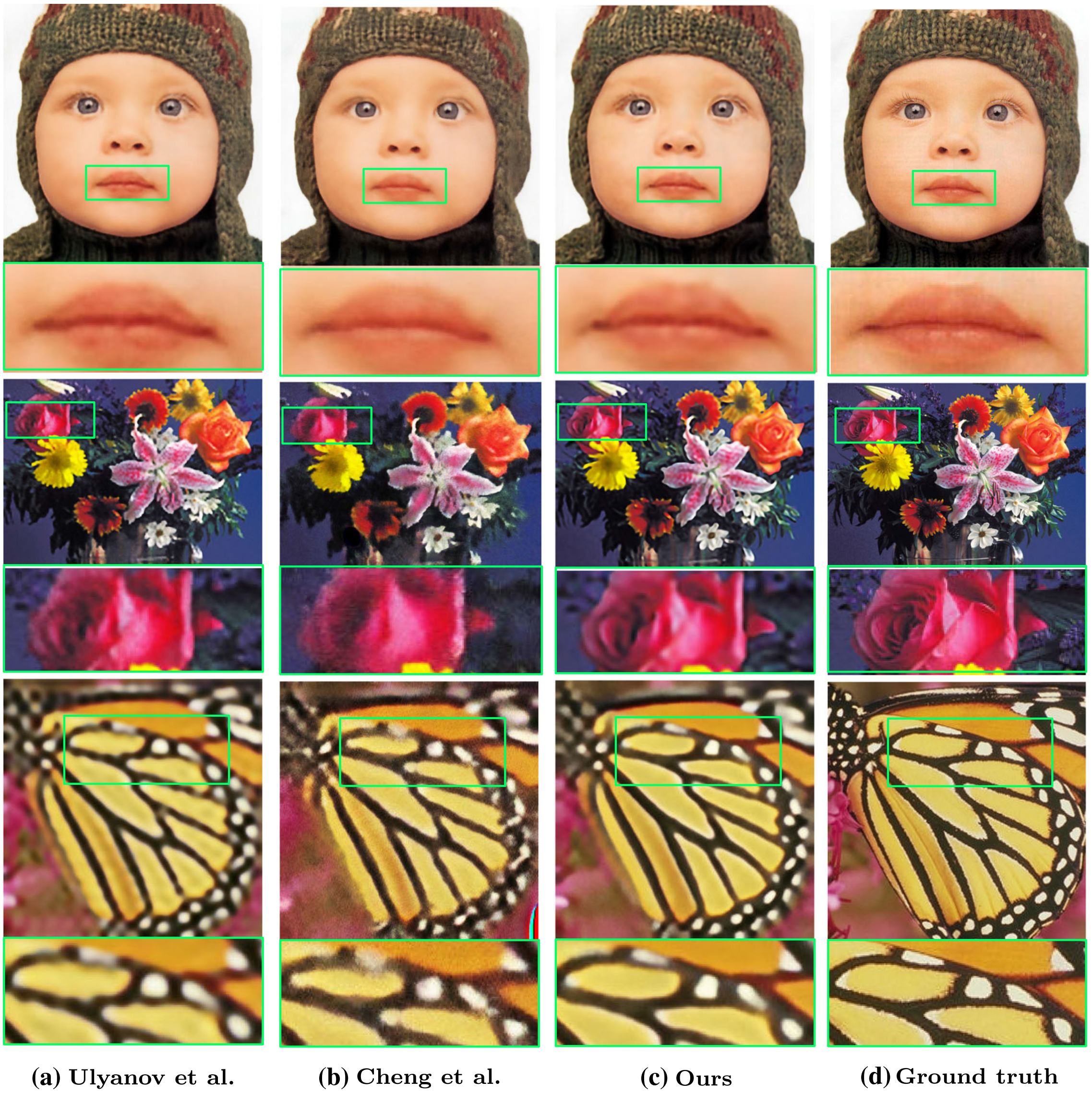
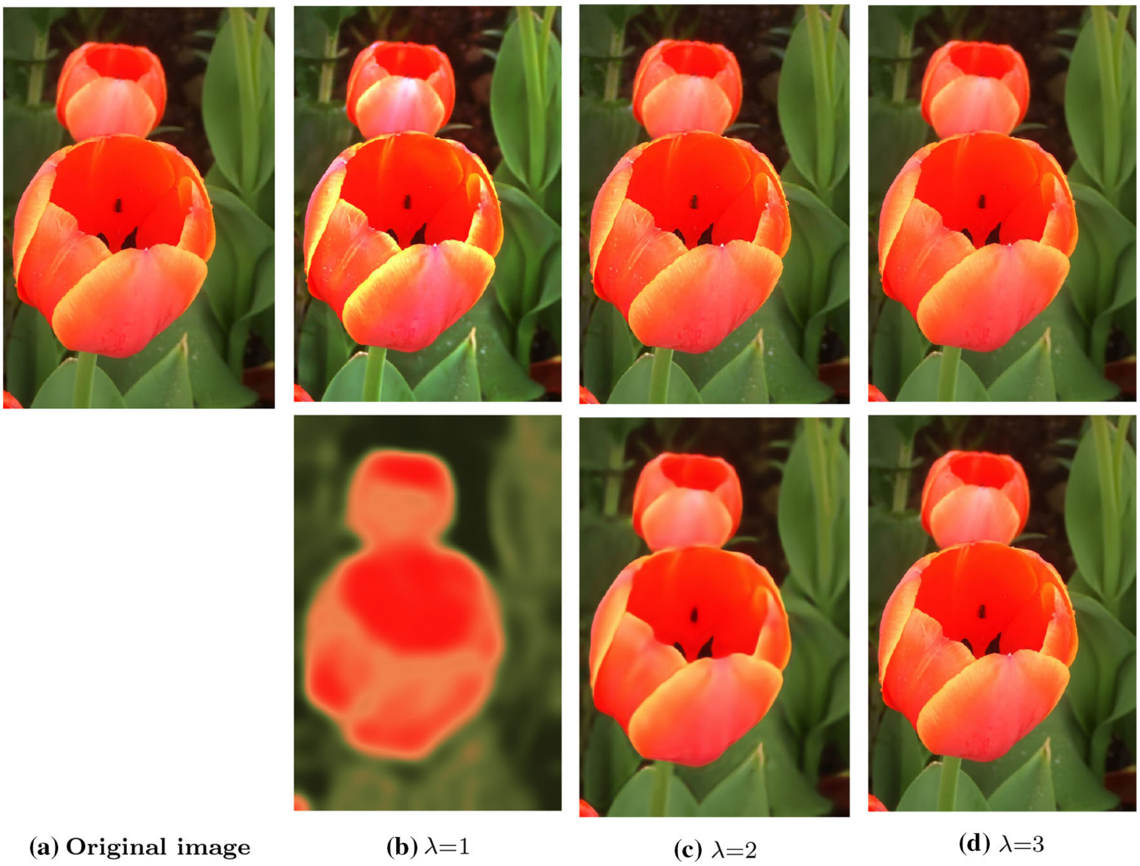


Fig. 15 **Super-resolution**. Results on the 'baby' image and the 'flowers' image for 4× super-resolution, and on the 'butterfly' image for 8× super-resolution. From the regions masked by the green rectangles, we observe our method is able to better recover details with fewer artifacts (best viewed digitally)

5.4 Super-resolution

In image super-resolution, a low-resolution image is given; the goal is to recover its scaled-up version. Following Ulyanov et al. (2018, 2020), the network generates a high-resolution image from the random noise input. The high-resolution image is then downsampled using a differentiable Lanczos filter to compute the loss with the provided low resolution image for optimizing the network. We report on the standard Set14 dataset by Zeyde et al. (2010) and Set5 by Bevilacqua et al. (2012). We evaluate the performance for an up-scaling of 4 and 8. For the super-resolution task, the deep image prior (Ulyanov et al. 2018, 2020) does not suffer from the performance degradation over iterations because the optimization objective strives to find the low-resolution image without high-frequency noise. Following Ulyanov et al. (2018, 2020), we report the PSNR score at a stopping iteration of 2,000 for the scaling of 4, and 4,000 for the scaling of 8. Results on Set 14 are provided in Table 4 and results on Set 5 are summarized in Table 5. On most images our method achieves better performance, not only for Ulyanov et al. (2018, 2020) but also compared to Heckel and Hand (2019) and Cheng et al. (2019). We provide a qualitative comparison in Fig. 15. We observe that our method produces fewer high-frequency artifacts than Ulyanov et al. (2018, 2020) and Cheng et al. (2019). We postulate that our Lipschitz normalization contributes to the benefit. Interestingly, our method also recovers fine details. A likely explanation is


(a) Original image

(b) λ=1

(c) λ=2

(d) λ=3

Fig. 16 **Image enhancement**. The goal is to enhance the image details. We obtain the smoothed images (second row) using the controlled deep image priors with different λ, as defined in Eq. (5). We then subtract the smoothed version from the original image to get fine details and enhance them (first row). The smaller the λ, the higher smoothness of the output images and the more enhancement to the image details

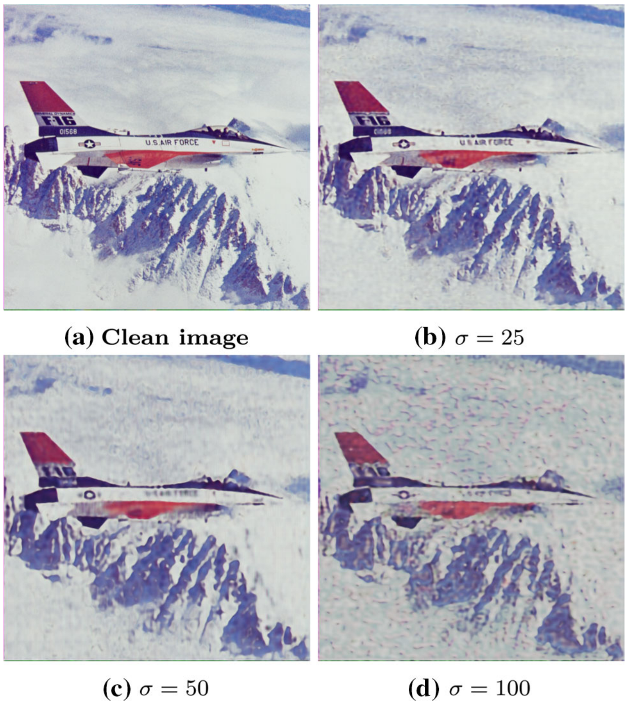
that our Gaussian upsampling is better at learning the desired higher frequencies. Note that fine details like textures are high-frequency compared to flat regions, but still relatively low-frequency compared to most artifacts.

5.5 Image Enhancement

Following Ulyanov et al. (2018, 2020), we also evaluate our method on image enhancement. The deep image prior performs sharpness enhancement by means of unsharp masking (Morishita et al. 1988; Shi et al. 2021), which can be described by $x_e = (x_0 - x_s) + x_0$, where an enhanced image is represented by x_e, an original image by x_0, an unsharp mask by $(x_0 - x_s)$ where x_s denotes the smoothed version of the original image. The smoothness of x_s controls the size of the region around the edge pixels that is affected by sharpening. The higher the smoothness, the wider the regions around the edges that got sharpened. The deep image prior obtains the smoothed images by stopping the optimization at different iterations. However, the smoothness of the output image is quite sensitive to the number of optimization iterations, which is hard to control. By contrast, our method is able to manipulate the smoothness of the output image by tuning λ in Eq. (5). Thus, we obtain the smoothed images with different λ, by optimizing the network in a fixed iteration of 5, 000. The smaller the λ, the higher the smoothness of the output images and the more enhancement to the image details, as shown in Fig. 16.

5.6 Success and Failure Cases

We return to the denoising task to analyze a success and failure case or our approach in Fig. 17. The goal is to remove additive Gaussian noise from a natural image. Our method performs well when the noise level is modest, as shown in Fig. 17b. However, with higher noise levels, the proposed method fails to remove the noise, as shown in Fig. 17d. We attribute this to the fact that in the frequency domain, additive Gaussian noise has equal intensity at different frequencies. By contrast, the power spectrum of a natural image decays rapidly from low frequencies to high frequencies (Ruderman 1994). Consequently, when the noise level is low, noise is


(a) Clean image

(b) $\sigma = 25$

(c) $\sigma = 50$

(d) $\sigma = 100$

Fig. 17 **Success and failure case** of our method for image denoising on image ‘F16’. Our method performs well when the noise level is modest (σ=25), while it fails to remove noise when the noise level is too high (σ=100)

usually dominant at high frequencies and the natural signal is more dominant at lower frequencies. However, the noise can also be more dominant at lower frequencies with higher level. In this case, separating low-frequencies from high-frequencies through spectral bias fails to remove the noise.

6 Conclusion

In this paper, we show the spectral bias leads inverse imaging networks to capture the deep image prior during optimization, independent of their architectures. We do so by introducing a metric, the Frequency Band Correspondence, which offers a spectral measurement of the deep image prior, revealing the low frequency natural image signals are learned faster and better than high-frequency noise signals. We also introduce Lipschitz normalization and Gaussian upsampling that allow to manipulate and adjust the spectral bias for inverse imaging problems. Besides these methods for controlling spectral bias, we further introduce a simple automatic stopping criterion to avoid superfluous computation. The experiments show that our method does not suffer from the performance degradation over iterations with controlled spectral bias and enables stopping the optimization automatically at an appropriate moment using the proposed stopping criterion. Our method also obtains favorable performance compared to current approaches for denoising, deblocking, inpainting, super-resolution and detail enhancement.

References

Arias, P., Facciolo, G., Caselles, V., & Sapiro, G. (2011). A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision, 93*(3), 319–347.

Arridge, S., Maass, P., Öktem, O., & Schönlieb, C. B. (2019). Solving inverse problems using data-driven models. *Acta Numerica, 28,* 1–174.

Asim, M., Shamshad, F., Ahmed, A. (2019). Patchdip exploiting patch redundancy in deep image prior for denoising. In: *NeurIPS Workshop on Solving Inverse Problems with Deep Networks*.

Bahrami, K., & Kot, A. C. (2014). A fast approach for no-reference image sharpness assessment based on maximum local variation. *IEEE Signal Processing Letters, 21*(6), 751–755.

Bertero, M., & Boccacci, P. (1998). *Introduction to inverse problems in imaging*. UK: IOP Publishing.

Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: *BMVC*.

Chakrabarty, P., Maji, S. (2019). The spectral bias of the deep image prior. In: *NeurIPS Workshop on Bayesian Deep Learning*.

Chen, D., Fan, Q., Liao, J., Aviles-Rivero, A., Yuan, L., Yu, N., & Hua, G. (2020). Controllable image processing via adaptive filterbank pyramid. *IEEE Transactions on Image Processing, 29,* 8043–8054.

Chen, Y., & Pock, T. (2016). Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1256–1272.

Chen, Y.C., Gao, C., Robb, E., Huang, J.B. (2020b). Nas-dip: Learning deep image prior with neural architecture search. In: *European Conference on Computer Vision (ECCV)*.

Cheng, Z., Gadelha, M., Maji, S., Sheldon, D. (2019). A bayesian perspective on the deep image prior. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Crete, F., Dolmiere, T., Ladret, P., Nicolas, M. (2007). The blur effect: Perception and estimation with a new no-reference perceptual blur metric. In: *SPIE HVEI*.

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing, 16*(8), 2080–2095.

Dai, T., Feng, Y., Wu, D., Chen, B., Lu, J., Jiang, Y., Xia, S.T. (2020). DIPDefend: Deep image prior driven defense against adversarial examples. In: *Proceedings of the 28th ACM International Conference on Multimedia*.

Daubechies, I., Defrise, M., & De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics, 57*(11), 1413–1457.

Dong, C., Deng, Y., Loy, C.C., Tang, X. (2015a). Compression artifacts reduction by a deep convolutional network. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 576–584.

Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(2), 295–307.

Dong, W., Shi, G., Ma, Y., & Li, X. (2015). Image restoration via simultaneous sparse coding: Where structured sparsity meets gaussian scale mixture. *International Journal of Computer Vision, 114*(2), 217–232.

Efros, A. A., & Leung, T. K. (1999). Texture synthesis by non-parametric sampling. *Proceedings of the Seventh IEEE International Conference on Computer Vision, 2,* 1033–1038.

Elad, M., Figueiredo, M. A., & Ma, Y. (2010). On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE, 98*(6), 972–982.

Engl, H. W., Hanke, M., & Neubauer, A. (1996). *Regularization of inverse problems* (Vol. 375). Berlin: Springer Science & Business Media.

Foi, A., Katkovnik, V., Egiazarian, K. (2006). Pointwise shape-adaptive dct for high-quality deblocking of compressed color images. In: *2006 14th European Signal Processing Conference*, pp. 1–5.

Gandelsman, Y., Shocher, A., Irani, M. (2019). double-dip: Unsupervised image decomposition via coupled deep-image-priors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hahn, J., Tai, X. C., Borok, S., & Bruckstein, A. M. (2011). Orientation-matching minimization for image denoising and inpainting. *International Journal of Computer Vision, 92*(3), 308–324.

He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Heckel, R., Hand, P. (2019). Deep decoder: Concise image representations from untrained non-convolutional networks. In: *International Conference on Learning Representations*.

Heckel, R., Soltanolkotabi, M. (2020). Denoising and regularization via exploiting the structural bias of convolutional generators. In: *International Conference on Learning Representations*.

Heide, F., Heidrich, W., Wetzstein, G. (2015). Fast and flexible convolutional sparse coding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ho, K., Gilbert, A., Jin, H., Collomosse, J. (2020). Neural architecture search for deep image prior. arXiv:2001.04776

Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*.

Jain, V., Seung, S. (2008). Natural image denoising with convolutional networks.

Jin, K. H., McCann, M. T., Froustey, E., & Unser, M. (2017). Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing, 26*(9), 4509–4522.

Katsaggelos, A. K. (1989). Iterative image restoration algorithms. *Optical Engineering, 28*(7), 287735.

Kattamis, A., Adel, T., Weller, A. (2019). Exploring properties of the deep image prior. In: *NeurIPS Workshop on Solving Inverse Problems with Deep Networks*.

Katznelson, Y. (2004). *An introduction to harmonic analysis*. Cambridge: Cambridge University Press.

Kindermann, S., Osher, S., & Jones, P. W. (2005). Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation, 4*(4), 1091–1115.

Kingma, D.P., Ba, J. (2015). Adam: A method for stochastic optimization. In: *International Conference on Learning Representations*.

Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2018). Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(11), 2599–2613.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W. (2017). 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.

Lefkimmiatis, S. (2018). Universal denoising networks: a novel cnn architecture for image denoising. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3204–3213.

Li, J., You, S., Robles-Kelly, A. (2018). A frequency domain neural network for fast image super-resolution. In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Lin, Z., He, J., Tang, X., & Tang, C. K. (2008). Limits of learning-based superresolution algorithms. *International Journal of Computer Vision, 80*(3), 406–420.

Liu, J., Sun, Y., Xu, X., Kamilov, U.S. (2019). Image restoration using total variation regularized deep image prior. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Lucas, A., Iliadis, M., Molina, R., & Katsaggelos, A. K. (2018). Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine, 35*(1), 20–36.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A. (2009). Non-local sparse models for image restoration. In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 2272–2279.

Mao, X., Shen, C., Yang, Y.B. (2016). Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: *NeurIPS*.

Mataev, G., Milanfar, P., Elad, M. (2019). Deepred: Deep image prior powered by red. In: *ICCV Workshop on Learning for Computational Imaging*.

McCann, M. T., Jin, K. H., & Unser, M. (2017). Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine, 34*(6), 85–95.

Miyato T, Kataoka T, Koyama M, Yoshida Y (2018) Spectral normalization for generative adversarial networks. In: *ICLR*.

Morishita, K., Yamagata, S., Okabe, T., Yokoyama, T., Hamatani, K. (1988). Unsharp masking for image enhancement. US Patent 4,794,531.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature, 381*(6583), 607–609.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A. (2016). Context encoders: feature learning by inpainting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544.

Portilla, J. (2009). Image restoration through l0 analysis-based sparse optimization in tight frames. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 3909–3912.

Protter, M., Elad, M., Takeda, H., & Milanfar, P. (2008). Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Transactions on Image Processing, 18*(1), 36–51.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A. (2019). On the spectral bias of neural networks. In: *International Conference on Machine Learning*.

Rasti, B., Koirala, B., Scheunders, P., Ghamisi, P. (2021). Undip: Hyperspectral unmixing using deep image prior. *IEEE Transactions on Geoscience and Remote Sensing*.

Ribes, A., & Schmitt, F. (2008). Linear inverse problems in imaging. *IEEE Signal Processing Magazine, 25*(4), 84–99.

Roth, S., & Black, M. J. (2009). Fields of experts. *International Journal of Computer Vision, 82*(2), 205.

Ruderman, D. L. (1994). The statistics of natural images. *Network: Computation in Neural Systems, 5*(4), 517–548.

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena, 60*(1–4), 259–268.

Sheikh, H. R., Sabir, M. F., & Bovik, A. C. (2006). A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing, 15*(11), 3440–3451.

Shi, Z., Chen, Y., Gavves, E., Mettes, P., & Snoek, C. G. (2021). Unsharp mask guided filtering. *IEEE Transactions on Image Processing, 30,* 7472–7485.

Simoncelli, E. P., & Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscience, 24*(1), 1193–1216.

Titterington, D. (1985). General structure of regularization procedures in image reconstruction. *Astronomy and Astrophysics, 144,* 381.

Ulyanov, D., Vedaldi, A., Lempitsky, V. (2018). Deep image prior. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ulyanov, D., Vedaldi, A., Lempitsky, V. (2020). Deep image prior. *International Journal of Computer Vision, 128*(7).

Vu, T., DiSpirito, A., Li, D., Wang, Z., Zhu, X., Chen, M., et al. (2021). Deep image prior for undersampling high-speed photoacoustic microscopy. *Photoacoustics, 22,* 100266.

Wan, Z., Zhang, B., Chen, D., Zhang, P., Chen, D., Liao, J., Wen, F. (2020). Bringing old photos back to life. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2747–2757.

Xu, Z.Q.J., Zhang, Y., Luo, T., Xiao, Y., Ma, Z. (2020). Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*.

Zeyde, R., Elad, M., Protter, M. (2010). On single image scale-up using sparse-representations. In: *International Conference on Curves and Surfaces*.

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing, 26*(7), 3142–3155.

Zhang, K., Zuo, W., & Zhang, L. (2018). Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing, 27*(9), 4608–4622.

Zukerman, J., Tirer, T., Giryes, R. (2020). Bp-dip: A backprojection based deep image prior. In: *2020 28th European Signal Processing Conference (EUSIPCO)*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.