Deep Image Prior

Dmitry Ulyanov
Skolkovo Institute of Science and Technology, Yandex
dmitry.ulyanov@skoltech.ru

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

Victor Lempitsky
Skolkovo Institute of Science and Technology
lempitsky@skoltech.ru

Abstract

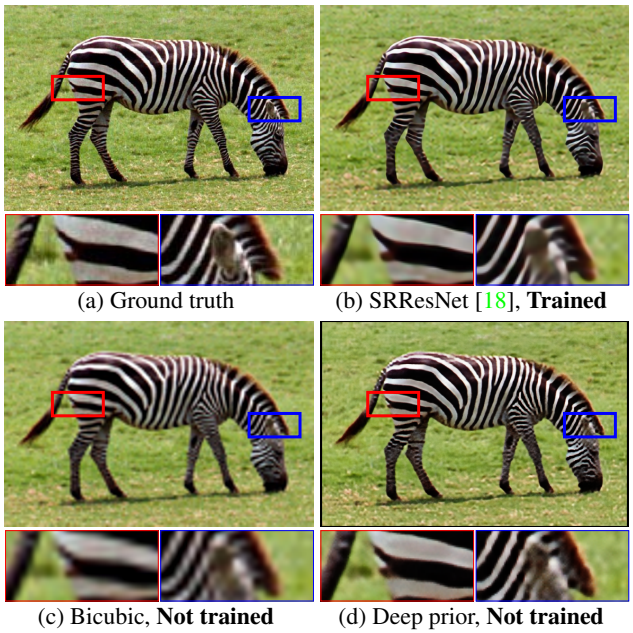
Deep convolutional networks have become a popular tool for image generation and restoration. Generally, their excellent performance is imputed to their ability to learn realistic image priors from a large number of example images. In this paper, we show that, on the contrary, the structure *of a generator network is sufficient to capture a great deal of low-level image statistics* prior to any learning. *In order to do so, we show that a randomly-initialized neural network can be used as a handcrafted prior with excellent results in standard inverse problems such as denoising, super-resolution, and inpainting. Furthermore, the same prior can be used to invert deep neural representations to diagnose them, and to restore images based on flash-no flash input pairs.*

Apart from its diverse applications, our approach highlights the inductive bias captured by standard generator network architectures. It also bridges the gap between two very popular families of image restoration methods: learning-based methods using deep convolutional networks and learning-free methods based on handcrafted image priors such as self-similarity.

1. Introduction

Deep convolutional neural networks (ConvNets) currently set the state-of-the-art in inverse image reconstruction problems such as denoising [4, 19] or single-image super-resolution [18, 28, 17]. ConvNets have also been used with great success in more "exotic" problems such as reconstructing an image from its activations within certain deep networks or from its HOG descriptor [7]. More generally, ConvNets with similar architectures are nowadays used to generate images using such approaches as generative adversarial networks [10], variational autoencoders [15], and direct pixelwise error minimization [8].

State-of-the-art ConvNets for image restoration and generation are almost invariably trained on large datasets of images. One may thus assume that their excellent performance



(a) Ground truth (b) SRResNet [18], **Trained**
(c) Bicubic, **Not trained** (d) Deep prior, **Not trained**

Figure 1: **Super-resolution using the deep image prior.** Our method uses a randomly-initialized convnet to upsample an image, using its strucrture as an image prior; similar to bicubic upsampling, this method does not require learning, but produces much cleaner results with sharper edges. In fact, our results are quite close to state-of-the-art super-resolution methods that use ConvNets learned from large datasets. The deep image prior works well for all inverse problems we could test.

Code and supplementary material are available at https://dmitryulyanov.github.io/deep_image_prior

is due to their ability to learn realistic image priors from data. However, learning alone is insufficient to explain the good performance of deep networks. For instance, the authors of [32] recently showed that the same image classification network that generalizes well when trained on genuine data can *also* overfit when presented with random labels. Thus, generalization requires the *structure* of the network to "resonate" with the structure of the data. However, the nature of this interaction remains unclear, particularly in the context of image generation.

In this work, we show that, contrary to expectations, a great deal of image statistics are captured by the *structure* of a convolutional image generator rather than by any learned capability. This is particularly true for the statistics required to solve various image restoration problems, where the image prior is required to integrate information lost in the degradation processes.

To show this, we apply *untrained* ConvNets to the solution of several such problems. Instead of following the common paradigm of training a ConvNet on a large dataset of example images, we fit a generator network to a single degraded image. In this scheme, the network weights serve as a parametrization of the restored image. The weights are randomly initialized and fitted to maximize their likelihood given a specific degraded image and a task-dependent observation model.

We show that this very simple formulation is very competitive for standard image processing problems such as denoising, inpainting and super-resolution. This is particularly remarkable because *no aspect of the network is learned from data*; instead, the weights of the network are always randomly initialized, so that the only prior information is in the structure of the network itself. To the best of our knowledge, this is the first study that directly investigates the prior captured by deep convolutional generative networks independently of learning the network parameters from images.

In addition to standard image restoration tasks, we show an application of our technique to understanding the information contained within the activations of deep neural networks. For this, we consider the "natural pre-image" technique of [20], whose goal is to characterize the invariants learned by a deep network by inverting it on the set of natural images. We show that an untrained deep convolutional generator can be used to replace the surrogate natural prior used in [20] (the TV norm) with dramatically improved results. Since the new regularizer, like the TV norm, is not learned from data but is entirely handcrafted, the resulting visualizations avoid potential biases arising form the use of powerful learned regularizers [7].

2. Method

Deep networks are applied to image generation by learning generator/decoder networks $x = f_\theta(z)$ that map a random code vector z to an image x. This approach can be used to sample realistic images from a random distribution [10]. Furthermore, the distribution can be conditioned on a corrupted observation x_0 to solve inverse problems such as denoising [4] and super-resolution [6].

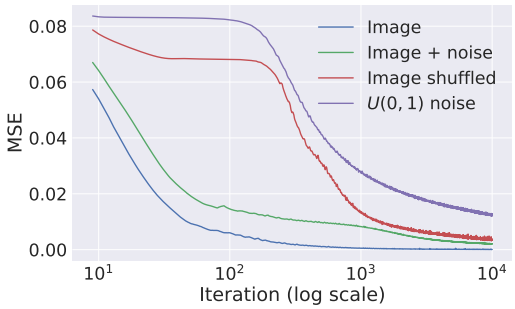


Figure 2: Learning curves for the reconstruction task using: a natural image, the same plus i.i.d. noise, the same randomly scrambled, and white noise. Naturally-looking images result in much faster convergence, whereas noise is rejected.

In this paper, we investigate the prior implicitly captured by the choice of a particular generator network structure, *before* any of its parameters are learned. We do so by interpreting the neural network as a *parametrization* $x = f_\theta(z)$ of an image $x \in \mathbb{R}^{3\times H\times W}$. Here $z \in \mathbb{R}^{C'\times H'\times W'}$ is a code tensor/vector and θ are the network parameters. The network itself alternates filtering operations such as convolution, upsampling and non-linear activation. In particular, most of our experiments are performed using am encoder-decorder "hourglass" architecture with as many as two million parameters θ (see Supplementary Material for details of all used architectures).

To demonstrate the power of this parametrization, we consider inverse tasks such as denoising, super-resolution and inpainting. These can be expressed as energy minimization problems of the type

$$x^* = \min_x E(x; x_0) + R(x), \tag{1}$$

where $E(x; x_0)$ is a task-dependent data term, x_0 the noisy/low-resolution/occluded image, and $R(x)$ a regularizer.

The choice of data term $E(x; x_0)$ is dictated by the application and will be discussed later. The choice of regularizer, which usually captures a generic prior on natural images, is more difficult and is the subject of much research. As a simple example, $R(x)$ may be the Total Variation (TV) of the image, which encourages solutions to contain uniform regions. In this work, we *replace* the regularizer $R(x)$ with the implicit prior captured by the neural network, as fol-

lows:

$$\theta^* = \operatorname*{argmin}_{\theta} E(f_\theta(z); x_0) \qquad x^* = f_{\theta^*}(z)\,. \tag{2}$$

The minimizer θ^* is obtained using an optimizer such as gradient descent starting from a *random initialization* of the parameters. Given a (local) minimizer θ^*, the result of the restoration process is obtained as $x^* = f_{\theta^*}(z)$. Note that it is also possible to optimize over the code z, but we usually initialize it randomly and keep it fixed.

In terms of (1), the prior $R(x)$ defined by (2) is an indicator function $R(x) = 0$ for all images that can be produced from z by a deep ConvNet of a certain architecture, and $R(x) = +\infty$ for all other signals. Since no aspect of the network is pre-trained from data, such *deep image prior* is effectively handcrafted, just like the TV norm. We show that this hand-crafted prior works very well for various image restoration tasks.

A parametrization with high noise impedance. One may wonder why a high-capacity network f_θ can be used as a prior at all. In fact, one may expect to be able to find parameters θ recovering any possible image x, including random noise, so that the network should not impose any restriction on the generated image. We now show that, while indeed almost any image can be fitted, the choice of network architecture has a major effect how the solution space is searched by methods such as gradient descent. In particular, we show that the network resists "bad" solutions and descends much more quickly towards naturally-looking images. The result is that minimizing (2) either results in a good-looking local optimum, or, at least, the optimization trajectory passes near one.

In order to study this effect quantitatively, we consider the most basic reconstruction problem: given a target image x_0, we want to find the value of the parameters θ^* that reproduce that image. This can be setup as the optimization of (2) using a data term comparing the generated image to x_0:

$$E(x; x_0) = \|x - x_0\|^2 \tag{3}$$

Plugging this in eq. (2) leads us to the optimization problem

$$\min_{\theta} \|f_\theta(z) - x_0\|^2 \tag{4}$$

Figure 2 shows the value of the energy $E(x; x_0)$ as a function of the gradient descent iterations for four different choices for the image x_0: 1) a natural image, 2) the same image plus additive noise, 3) the same image after randomly permuting the pixels, and 4) white noise. It is apparent from the figure that optimization is much faster for cases 1) and 2), whereas the parametrization presents significant "inertia" for cases 3) and 4).

Thus, although in the limit the parametrization *can* fit unstructured noise, it does so very reluctantly. In other words, the parametrization offers high impedance to noise and low impedance to signal. Therefore for most applications, we restrict the number of iterations in the optimization process (2) to a certain number of iterations. The resulting prior then corresponds to projection onto a reduced set of images that can be produced from z by ConvNets with parameters θ that are not too far from the random initialization θ_0.

3. Applications

We now show experimentally how the proposed prior works for diverse image reconstruction problems. Due to space limitations, our evaluation in the main text is restricted to few examples and numbers. The reader is therefore strongly encouraged to address the **Supplementary material** [29] for more extensive evaluation and for extra details.

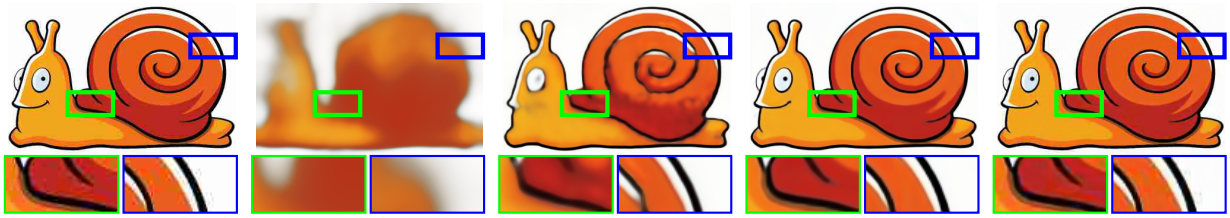
Denoising and generic reconstruction. As our parametrization presents high impedance to image noise, it can be naturally used to filter out noise from an image. The aim of denoising is to recover a clean image x from a noisy observation x_0. Sometimes the degradation model is known: $x_0 = x + \epsilon$ where ϵ follows a particular distribution. However, more often in *blind denoising* the noise model is unknown.

Here we work under the blindness assumption, but the method can be easily modified to incorporate information about noise model. We use the same exact formulation as eqs. (3) and (4) and, given a noisy image x_0, recover a clean image $x^* = f_{\theta^*}(z)$ after substituting the minimizer θ^* of eq. (4).

Our approach does not require a model for the image degradation process that it needs to revert. This allows it to be applied in a "plug-and-play" fashion to image restoration tasks, where the degradation process is complex and/or unknown and where obtaining realistic data for supervised training is highly problematic. We demonstrate this capability by several qualitative examples in fig. 4 and in the supplementary material, where our approach uses the quadratic energy (3) leading to formulation (4) to restore images degraded by complex and unknown compression artifacts. Figure 3 (top row) also demonstrates the applicability of the method beyond natural images (cartoon images in this case).

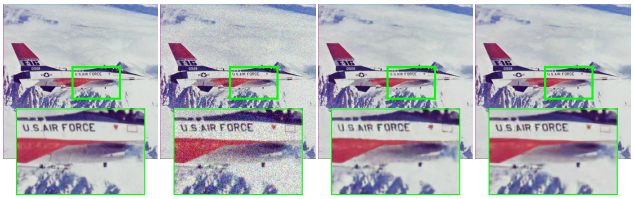
We evaluate our denoising approach on the standard dataset[1], consisting of 9 colored images with noise strength of $\sigma = 25$. We achieve a PSNR of 29.22 after 1800 optimization steps. The score is improved up to 30.43 if we additionally average the restored images obtained in the last iterations (using exponential sliding window). If averaged over two optimization runs our method further improves up

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



Corrupted | 100 iterations | 600 iterations | 2400 iterations | 50K iterations

Figure 3: **Blind restoration of a JPEG-compressed image.** (*electronic zoom-in recommended*) Our approach can restore an image with a complex degradation (JPEG compression in this case). As the optimization process progresses, the deep image prior allows to recover most of the signal while getting rid of halos and blockiness (after 2400 iterations) before eventually overfitting to the input (at 50K iterations).



(a) GT (b) Input (c) Ours (d) CBM3D

Figure 4: **Blind image denoising.** The deep image prior is successful at recovering both man-made and natural patterns. For reference, the result of a state-of-the-art non-learned denoising approach [5] is shown.

to 31.00 PSNR. For the reference, the scores for the two popular approaches CMB3D [5] and Non-local means [3] that do not require pretraining are 31.42 and 30.26 respectively.

Super-resolution. The goal of super-resolution is to take a low resolution (LR) image $x_0 \in \mathbb{R}^{3\times H\times W}$ and upsampling factor t, and generate a corresponding high resolution (HR) version $x \in \mathbb{R}^{3\times tH\times tW}$. To solve this inverse problem, the data term in (2) is set to:

$$E(x;x_0) = \|d(x) - x_0\|^2 \tag{5}$$

where $d(\cdot) : \mathbb{R}^{3\times tH\times tW} \to \mathbb{R}^{3\times H\times W}$ is a *downsampling operator* that decimates an image by a factor t. Hence, the problem is to find the HR image x that, when downsampled, is the same as the LR image x_0.

Super-resolution is an ill-posed problem because there are infinitely many HR images x that reduce to the same LR image x_0 (i.e. the operator d is far from surjective). Regularization is required in order to select, among the infinite minimizers of (5), the most plausible ones.

Following eq. (2), we regularize the problem by considering the reparametrization $x = f_\theta(z)$ and optimizing the resulting energy w.r.t. θ. Optimization still uses gradient descent, exploiting the fact that both the neural network and the most common downsampling operators, such as Lanczos, are differentiable.

We evaluate super-resolution ability of our approach using Set5 [2] and Set14 [31] datasets. We use a scaling factor of $4\times$ to compare to other works, while we show results with other scaling factors in [29]. We fix the number of optimization steps to be constant for every image.

Qualitative comparison with bicubic upsampling and state-of-the art learning-based methods SRResNet [18], LapSRN [28] is presented in fig. 5. Our method can be fairly compared to bicubic, as both methods never use other data than a given low-resolution image. Visually, we approach the quality of leaning-based methods that use the MSE loss. GAN-based [10] methods SRGAN [18] and EnhanceNet [27] (not shown in the comparison) intelligently hallucinate fine details of the image, which is impossible with our method that uses absolutely no information about the world of HR images.

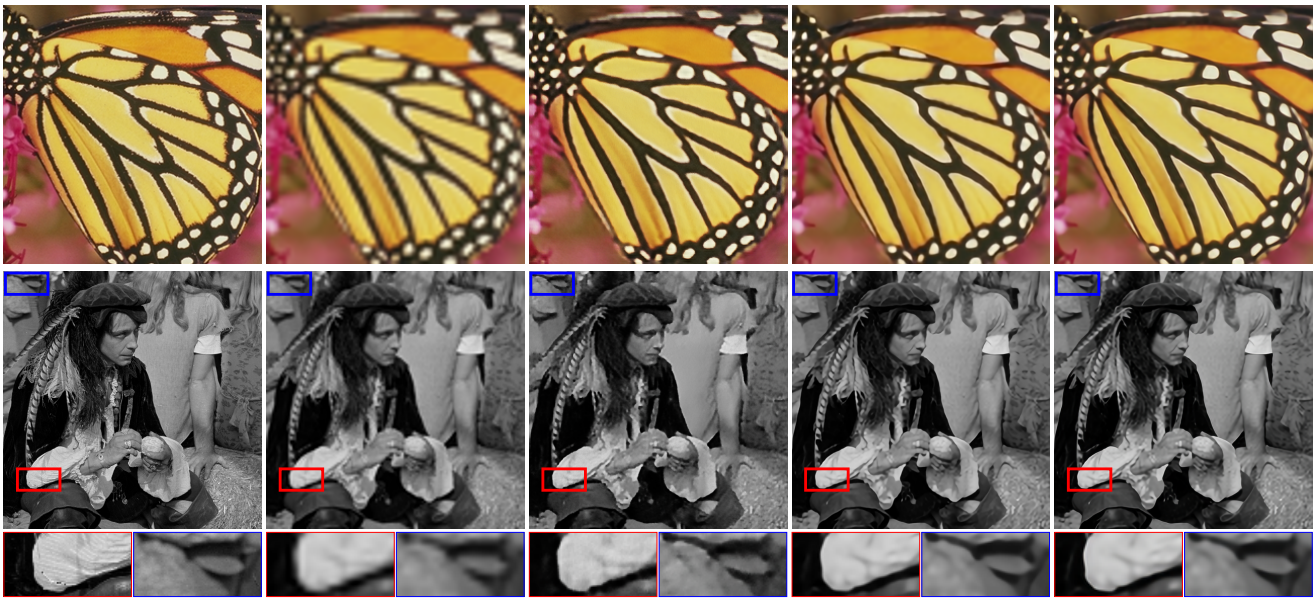
We compute PSNRs using center crops of the generated images. Our method achieves 27.95 and 35.06 PSNR on Set5 and Set14 datasets respectively. Bicubic upsampling gets a lower score of 26.70 and 33.78, while SRResNet has PSNR of 30.09 and 37.23. While our method is still outperformed by learning-based approaches, it does considerably better than bicubic upsampling. Visually, it seems to close most of the gap between bicubic and state-of-the-art trained ConvNets (c.f. fig. 1,fig. 5 and [29]).

Inpainting. In image inpainting, one is given an image x_0 with missing pixels in correspondence of a binary mask $m \in \{0,1\}^{H\times W}$; the goal is to reconstruct the missing data. The corresponding data term is given by

$$E(x;x_0) = \|(x - x_0)\odot m\|^2, \tag{6}$$

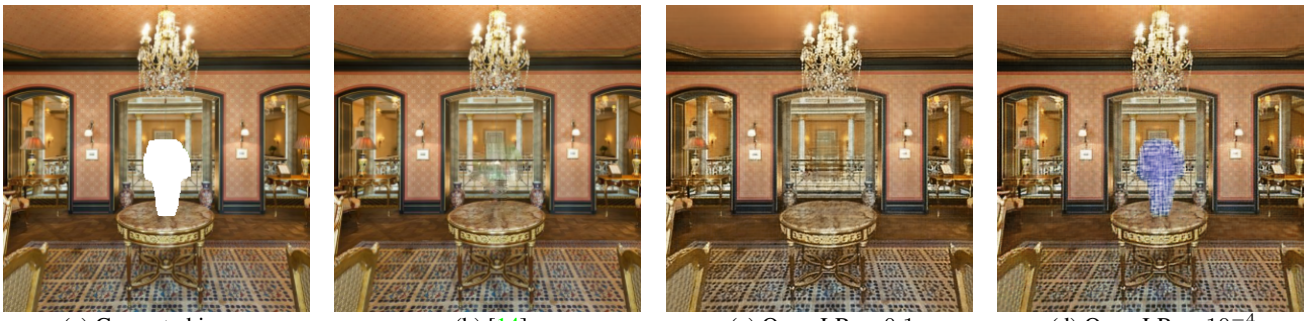
where $\odot$ is Hadamard's product. The necessity of a data prior is obvious as this energy is independent of the values of the missing pixels, which would therefore never change after initialization if the objective was optimized directly over pixel values x. As before, the prior is introduced by optimizing the data term w.r.t. the reparametrization (2).

In the first example (fig. 7, top row) inpainting is used to remove text overlaid on an image. Our approach is com-



(a) Original image (b) Bicubic, **Not trained** (c) Ours, **Not trained** (d) LapSRN, **Trained** (e) SRResNet, **Trained**

Figure 5: **4x image super-resolution.** Similarly to e.g. bicubic upsampling, our method never has access to any data other than a single low-resolution image, and yet it produces much cleaner results with sharp edges close to state-of-the-art super-resolution methods (LapSRN [17], SRResNet [18]) which utilize networks trained from large datasets.



(a) Corrupted image (b) [14] (c) Ours, LR $= 0.1$ (d) Ours, LR $= 10^{-4}$

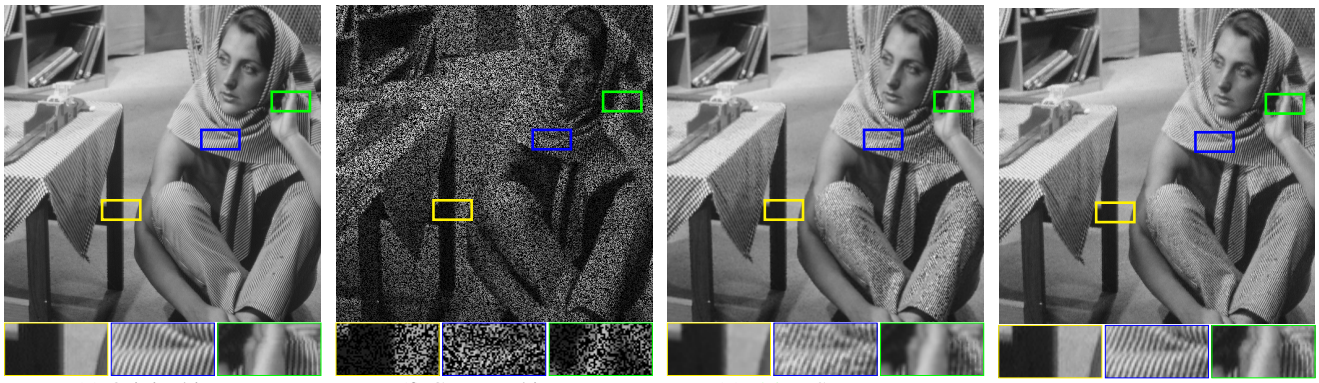
Figure 6: **Region inpainting.** Our method is able to successfully inpaint large regions. Despite using no learning, results are comparable to [14] which does. The choice of hyper-parameters is important (for example (d) demonstrates sensitivity to the learning rate), but a good setting works well for all images.

pared to the method of [26] specifically designed for inpainting. We observe almost perfectly transparent inpainting, while for [26] the text mask remains visible in some regions.

Next, fig. 7 (bottom) considers inpainting with masks randomly sampled according to a binary Bernoulli distribution. First, a mask is sampled to drop 50% of pixels at random. We compare our approach to a method of [24] based on convolutional sparse coding. To obtain results for [24] we first decompose the corrupted image x_0 into low and high frequency components similarly to [11] and run their method on the high frequency part. For a fair comparison we use the version of their method, where a dictionary is built using the input image (shown to perform better in [24]). The quantitative comparison on the standard data set [13] for our method is given in table 1, showing a strong quantitative advantage of the proposed approach compared to convolutional sparse coding. In fig. 7 (bottom) we present a representative qualitative visual comparison with [24].

We also apply our method to inpainting of large holes. Being non-trainable, our method is not expected to work correctly for "highly-semantical" large-hole inpainting (e.g. face inpainting). Yet, it works surprisingly well for other situations. We compare to a learning-based method of [14] in fig. 6. The deep image prior utilizes context of the image and interpolates the unknown region with textures from the

(a) Original image (b) Corrupted image (c) Shepard networks [26] (d) Deep Image Prior



(e) Original image (f) Corrupted image (g) [24], PSNR = 28.1 (h) Deep Img. Prior, PSNR = 30.9

Figure 7: **Comparison with two recent inpainting approaches**. Top – comparison with Shepard networks [26] on text inpainting example. Bottom – comparison with convolutional sparse coding [24] on inpainting 50% of missing pixels. In both cases, our approach performs better on the images used in the respective papers.

	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage
Papyan et al.	28.14	31.44	34.58	35.04	29.91	27.90	31.18	31.34	32.35	31.92	28.05
Ours	**30.88**	**32.84**	**37.52**	**36.05**	**31.22**	**28.83**	**32.43**	**34.40**	**33.15**	**32.26**	**29.85**

Table 1: Comparison between our method and the algorithm in [24]. See fig. 7 bottom row for visual comparison.

known part. Such behaviour highlights the relation between the deep image prior and traditional self-similarity priors.

In fig. 8, we compare deep priors corresponding to several architectures. Our findings here (and in other similar comparisons) seem to suggest that having deeper architecture is beneficial, and that having skip-connections that work so well for recognition tasks (such as semantic segmentation) is highly detrimental.

Natural pre-image. The natural pre-image method of [20] is a *diagnostic* tool to study the invariances of a lossy function, such as a deep network, that operates on natural images. Let Φ be the first several layers of a neural network trained to perform, say, image classification. The pre-image is the set $\Phi^{-1}(\Phi(x_0)) = \{x \in \mathcal{X} : \Phi(x) = \Phi(x_0)\}$ of images that result in the *same representation* $\Phi(x_0)$. Looking at this set reveals which information is lost by the network, and which invariances are gained.

Finding pre-image points can be formulated as minimizing the data term $E(x; x_0) = \|\Phi(x) - \Phi(x_0)\|^2$. However, optimizing this function directly may find "artifacts", i.e. non-natural images for which the behavior of the network Φ is in principle unspecified and that can thus drive it arbitrarily. More meaningful visualization can be obtained by restricting the pre-image to a set $\mathcal{X}$ of natural images, called a *natural pre-image* in [20].

In practice, finding points in the natural pre-image can be done by regularizing the data term similarly to the other inverse problems seen above. The authors of [20] prefer to use the TV norm, which is a weak natural image prior, but is relatively unbiased. On the contrary, papers such as [7] learn to invert a neural network from examples, resulting in better looking reconstructions, which however may be

(a) Input (white=masked) (b) Encoder-decoder, depth=6 (c) Encoder-decoder, depth=4

(d) Encoder-decoder, depth=2 (e) ResNet, depth=8 (f) U-net, depth=5

Figure 8: **Inpainting using different depths and architectures.** The figure shows that much better inpainting results can be obtained by using deeper random networks. However, adding skip connections to ResNet in U-Net is highly detrimental.

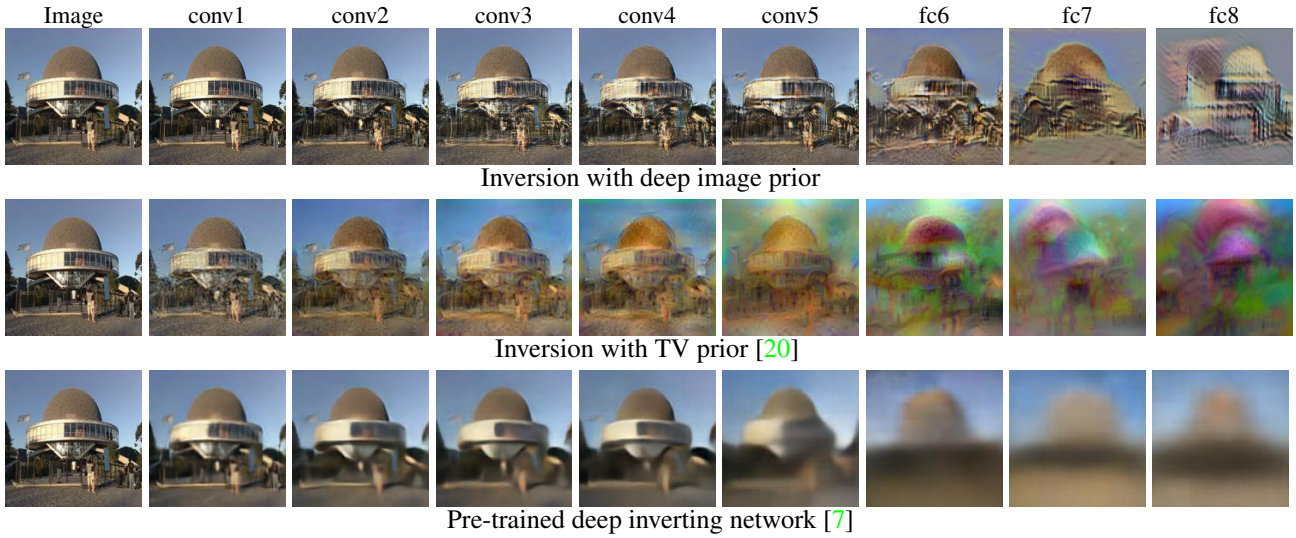


Figure 9: **AlexNet inversion.** Given the image on the left, we show the natural pre-image obtained by inverting different layers of AlexNet (trained for classification on ImageNet ISLVRC) using three different regularizers: the Deep Image prior, the TV norm prior of [20], and the network trained to invert representations on a hold-out set [7]. The reconstructions obtained with the deep image prior are in many ways at least as good as [7], yet they are not biased by the learning process.

biased towards the learn data-driven inversion prior. Here, we propose to use the deep image prior (2) instead. As this is handcrafted like the TV-norm, it is not biased towards a particular training set. On the other hand, it results in inversions at least as interpretable as the ones of [7].

For evaluation, our method is compared to the ones of [21] and [7]. Figure 9 shows the results of inverting representations Φ obtained by considering progressively deeper subsets of AlexNet [16]: `conv1`, `conv2`, ..., `conv5`, `fc6`, `fc7`, and `fc8`. Pre-images are found either by optimizing (2) using a structured prior.

As seen in fig. 9, our method results in dramatically improved image clarity compared to the simple TV-norm. The difference is particularly remarkable for deeper layers

such as `fc6` and `fc7`, where the TV norm still produces noisy images, whereas the structured regularizer produces images that are often still interpretable. Our approach also produces more informative inversions than a learned prior of [7], which have a clear tendency to regress to the mean.

Flash-no flash reconstruction. While in this work we focus on single image restoration, the proposed approach can be extended to the tasks of the restoration of multiple images, e.g. for the task of video restoration. We therefore conclude the set of application examples with a qualitative example demonstrating how the method can be applied to perform restoration based on pairs of images. In particular, we consider flash-no flash image pair-based restoration [25], where the goal is to obtain an image of a scene with the lighting similar to a no-flash image, while using the flash image as a guide to reduce the noise level.

In general, extending the method to more than one image is likely to involve some coordinated optimization over the input codes z that for single-image tasks in our approach was most often kept fixed and random. In the case of flash-no-flash restoration, we found that good restorations were obtained by using the denoising formulation (4), while using flash image as an input (in place of the random vector z). The resulting approach can be seen as a non-linear generalization of guided image filtering [12]. The results of the restoration are given in the fig. 10.

4. Related work

Our method is obviously related to image restoration and synthesis methods based on learnable ConvNets and referenced above. At the same time, it is as much related to an alternative group of restoration methods that avoid training on the hold-out set. This group includes methods based on joint modeling of groups of similar patches inside corrupted image [3, 5, 9], which are particularly useful when the corruption process is complex and highly variable (e.g. spatially-varying blur [1]). Also in this group are methods based on fitting dictionaries to the patches of the corrupted image [22, 31] as well as methods based on convolutional sparse coding [30], which can also fit statistical models similar to shallow ConvNets to the reconstructed image [24]. The work [19] investigates the model that combines ConvNet with a self-similarity based denoising and thus also bridges the two groups of methods, but still requires training on a hold-out set.

Overall, the prior imposed by deep ConvNets and investigated in this work seems to be highly related to self-similarity-based and dictionary-based priors. Indeed, as the weights of the convolutional filters are shared across the entire spatial extent of the image this ensures a degree of self-similarity of individual patches that a generative ConvNet can potentially produce. The connections between ConvNets and convolutional sparse coding run even deeper and are investigated in [23] in the context of recognition networks, and more recently in [24], where a single-layer convolutional sparse coding is proposed for reconstruction tasks. The comparison of our approach with [24] (fig. 7 and table 1) however suggests that using deep ConvNet architectures popular in modern deep learning-based approaches may lead to more accurate restoration results at least in some circumstances.

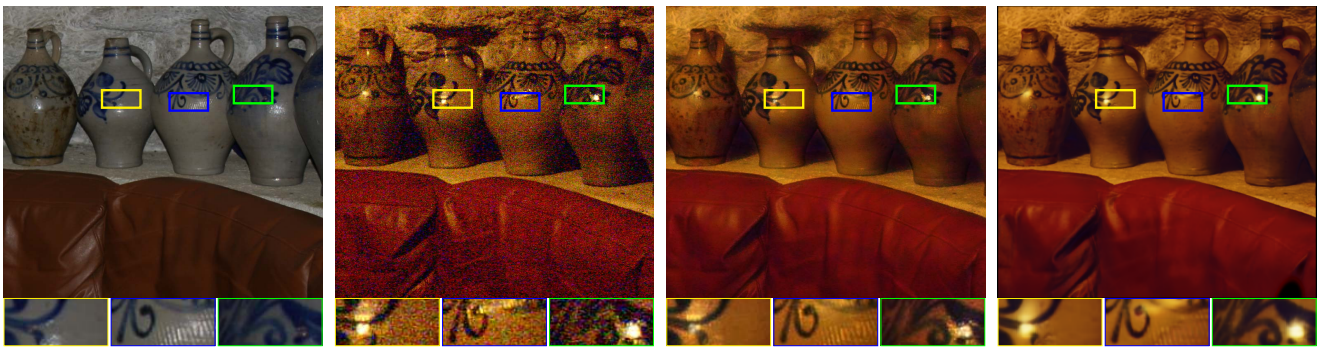
5. Discussion

We have investigated the role of the convolutional network architecture in the success of recent ConvNet-based image restoration methods. We have teased apart the contribution of the prior imposed by this architecture from the contribution of the information transfered from external images through learning. Along the way, we have shown that a simple approach of fitting randomly-initialized ConvNets to corrupted images works as a "Swiss knife" for restoration problems. The use of this "Swiss knife" does not require modeling of the degradation process or pre-training. Admittedly, the approach is computationally heavy (taking several minutes of GPU computation for 512x512 image).

In many ways, our results go against the common narrative that attributes the recent successes of deep learning-based methods in imaging to the shift from using hand-crafted priors to learning everything from data. It turns out that much of the success can be also attributed to switching from worse hand-crafted priors to better hand-crafted priors (hidden inside learnable deep ConvNets). This validates the importance of developing new deep learning architectures.

Acknowledgements

DU and VL are supported by the Ministry of Education and Science of the Russian Federation (grant 14.756.31.0001) and AV is supported by ERC 677195-IDIU.



(a) Flash (b) No flash (c) Joint bilateral [25] (d) Deep image prior

Figure 10: **Reconstruction based on flash and no-flash image pair.** The deep image prior allows to obtain low-noise reconstruction with the lighting very close to the no-flash image. It is more successful at avoiding "leaks" of the lighting patterns from the flash pair than joint bilateral filtering [25] (c.f. blue inset).

References

[1] Y. Bahat, N. Efrat, and M. Irani. Non-uniform blind deblurring by reblurring. In *Proc. CVPR*, pages 3286–3294, 2017. 8

[2] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proc. BMVC*, pages 1–10, 2012. 4

[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. CVPR*, volume 2, pages 60–65. IEEE, 2005. 4, 8

[4] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *Proc. CVPR*, pages 2392–2399, 2012. 1, 2

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 4, 8

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proc. ECCV*, pages 184–199, 2014. 2

[7] A. Dosovitskiy and T. Brox. Inverting convolutional networks with convolutional networks. In *Proc. CVPR*, 2016. 1, 2, 6, 7, 8

[8] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proc. CVPR*, pages 1538–1546, 2015. 1

[9] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proc. ICCV*, pages 349–356, 2009. 8

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014. 1, 2, 4

[11] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution. In *ICCV*, pages 1823–1831. IEEE Computer Society, 2015. 5

[12] K. He, J. Sun, and X. Tang. Guided image filtering. *T-PAMI*, 35(6):1397–1409, 2013. 8

[13] F. Heide, W. Heidrich, and G. Wetzstein. Fast and flexible convolutional sparse coding. In *Proc. CVPR*, pages 5135–5143, 2015. 5

[14] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 36(4):107:1–107:14, 2017. 5

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014. 1

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 7

[17] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 5

[18] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 4, 5

[19] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proc. CVPR*, 2016. 1, 8

[20] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proc. CVPR*, 2015. 2, 6, 7

[21] A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *IJCV*, 2016. 7

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010. 8

[23] V. Papyan, Y. Romano, and M. Elad. Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18(83):1–52, 2017. 8

[24] V. Papyan, Y. Romano, J. Sulam, and M. Elad. Convolutional dictionary learning via local processing. In *Proc. ICCV*, 2017. 5, 6, 8

[25] G. Petschnigg, R. Szeliski, M. Agrawala, M. F. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004. 8, 9

[26] J. S. J. Ren, L. Xu, Q. Yan, and W. Sun. Shepard convolutional neural networks. In *Proc. NIPS*, pages 901–909, 2015. 5, 6

[27] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 4

[28] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 4

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Supplementary material. `https://dmitryulyanov.github.io/deep_image_prior`. 3, 4

[30] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Proc. CVPR*, pages 2528–2535, 2010. 8

[31] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, volume 6920 of *Lecture Notes in Computer Science*, pages 711–730. Springer, 2010. 4, 8

[32] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *Proc. ICLR*, 2017. 2