IJCV manuscript No.
(will be inserted by the editor)

Deep Image Prior

Dmitry Ulyanov · Andrea Vedaldi · Victor Lempitsky

Received: date / Accepted: date

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.

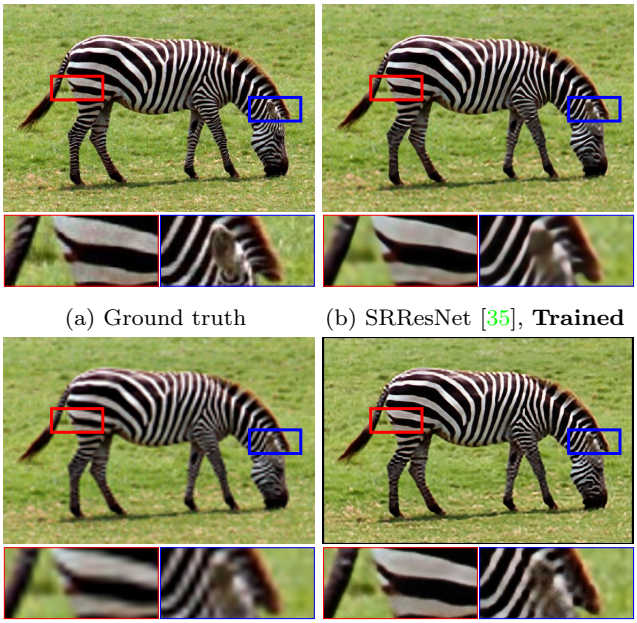
Apart from its diverse applications, our approach highlights the inductive bias captured by standard generator network architectures. It also bridges the gap between two very popular families of image restoration methods: learning-based methods using deep convolutional networks and learning-free methods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

D. Ulyanov
Skolkovo Institute of Science and Technology
E-mail: dmitry.ulyanov@skoltech.ru

A. Vedaldi
Oxford University
E-mail: vedaldi@robots.ox.ac.uk

V. Lempitsky
Skolkovo Institute of Science and Technology
E-mail: lempitsky@skoltech.ru

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



(a) Ground truth (b) SRResNet [35], **Trained**

(c) Bicubic, **Not trained** (d) Deep prior, **Not trained**

Fig. 1: **Super-resolution using the deep image prior.** Our method uses a randomly-initialized ConvNet to upsample an image, using its structure 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.

Keywords Convolutional networks, generative deep networks, inverse problems, image restoration, image superresolution, image denoising, natural image prior.

1 Introduction

Deep convolutional neural networks (ConvNets) currently set the state-of-the-art in inverse image reconstruction problems such as denoising [10,36] or single-image super-resolution [35,52,33]. 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 [14]. More generally, ConvNets with similar architectures are nowadays used to generate images using such approaches as generative adversarial networks [19], variational autoencoders [31], and direct pixelwise error minimization [15,5].

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 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 [59] 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 the belief that learning is necessary for building useful image priors, a great deal of image statistics are captured by the *structure* of a convolutional image generator independent of learning. 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 (fig. 1). 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.

Stated in a different way, we cast reconstruction as a *conditional* image generation problem and show that the only information required to solve it is contained in the single degraded input image *and* the handcrafted structure of the network used for reconstruction.

We show that this very simple formulation is very competitive for standard image processing problems such as denoising, inpainting, super-resolution, and detail enhancement. 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 [37], 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 [37] (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 [14]. Likewise, we show that the same regularizer works well for "activation maximization", namely the problem of synthesizing images that highly activate a certain neuron [16].

2 Method

A deep generator network can be thought of as a parametric function $x = f_\theta(z)$ that maps a code vector z to an image x. Such as generator is often used to model a complex distribution $p(x)$ over the images as a transformation of a simple distribution over the codes, such as a Gaussian [19]. The normal expectation is that the network parameters must be learned from data in order for the network to acquire useful information about the images. Instead, we show here that a significant amount of information is already contained in the *structure* of f_θ before any training is performed.

We do so by interpreting the neural network as a *parametrization* $x = f_\theta(z)$ of the image $x \in \mathbb{R}^{3\times H\times W}$. In this view, the code is a fixed random tensor $z \in \mathbb{R}^{C'\times H'\times W'}$ and the network maps the parameters θ, comprising the weights and bias of the filters in the network, to the image x. The network itself has a standard structure and alternates filtering operations such as linear convolution, upsampling and non-linear activation functions.

Without training, we cannot expect the a network f_θ to know about high-level concepts such as the appearance of human faces or animals. However, a we demonstrate, the network does contain knowledge about

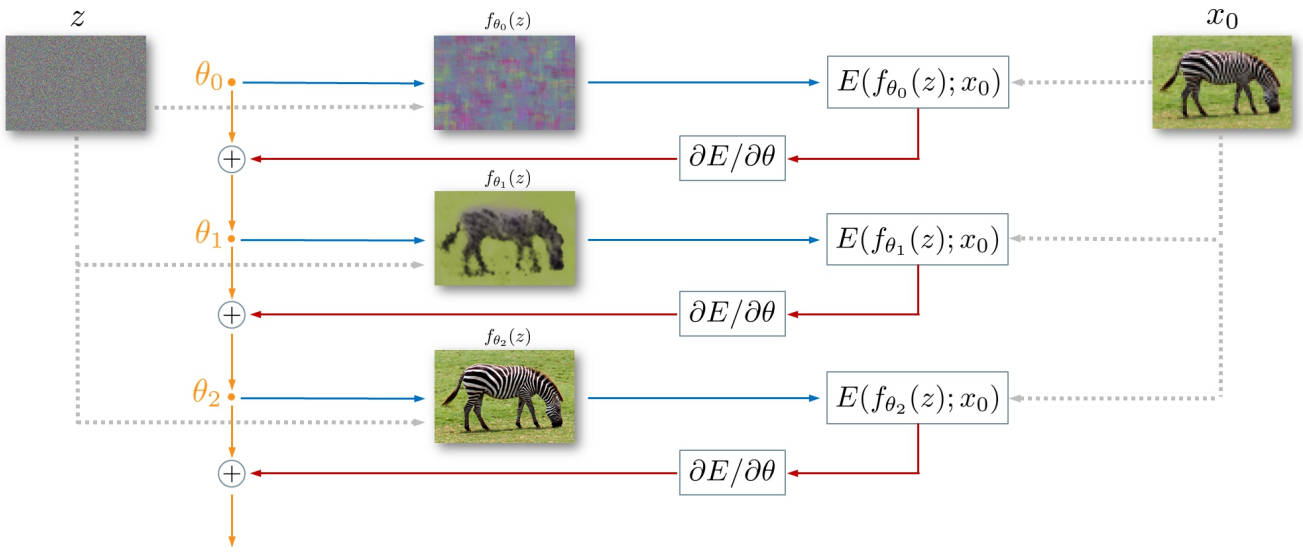


Fig. 2: **Image restoration using the deep image prior.** Starting from a random weights θ_0, we iteratively update them in order to minimize the data term eq. (2). At every iteration the weights θ are mapped to an image $x = f_\theta(z)$, where z is a fixed tensor and the mapping f is a neural network with parameters θ. The image x is used to compute the task-dependent loss $E(x, x_0)$. The gradient of the loss w.r.t. the weights θ is then computed and used to update the parameters.

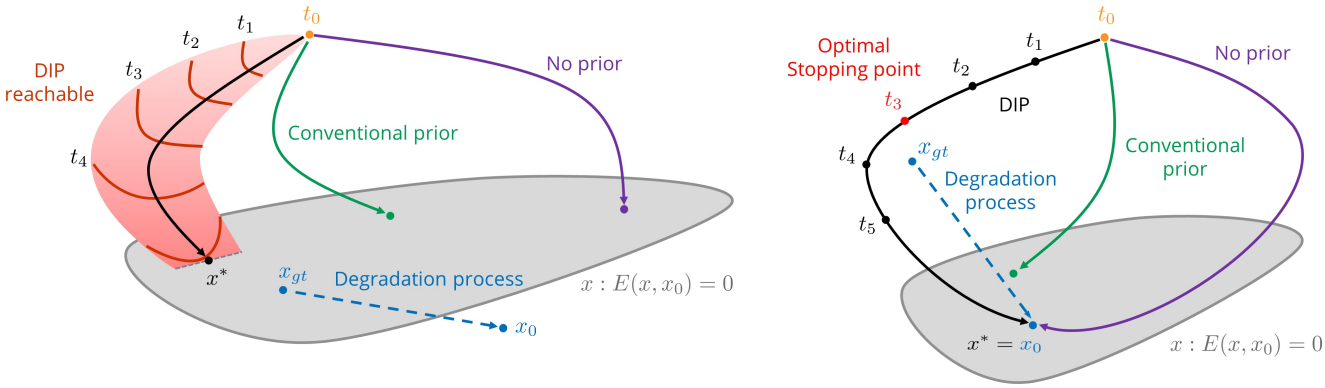


Fig. 3: **Restoration with priors – image space visualization.** We consider the problem of reconstructing an image x_{gt} from a degraded measurement x_0. We distinguish two cases. **Left** – in the first case, exemplified by super-resolution, the ground-truth solution x_{gt} belongs to a manifold of points x that have null energy $x : E(x, x_0) = 0$ (shown in gray) and optimization can land on a point x^* still quite far from x_{gt} (purple curve). Adding a conventional prior $R(x)$ tweaks the energy so that the optimizer x^* is closer to the ground truth (green curve). The deep image prior has a similar effect, but achieves it by tweaking the optimization trajectory via reparametrization, often with better results than conventional priors. **Right** – in the second case, exemplified by denoising, the ground truth x_{gt} has non-zero cost $E(x_{\text{gt}}, x_0) > 0$. Here, if run for long enough, fitting with deep image prior will obtain a solution with near zero cost quite far from x_{gt}. However, often the optimization path will pass close to x_{gt}, and an early stopping (here at time t_3) will recover good solution. Below, we show that deep image prior often helps for problems of both types.

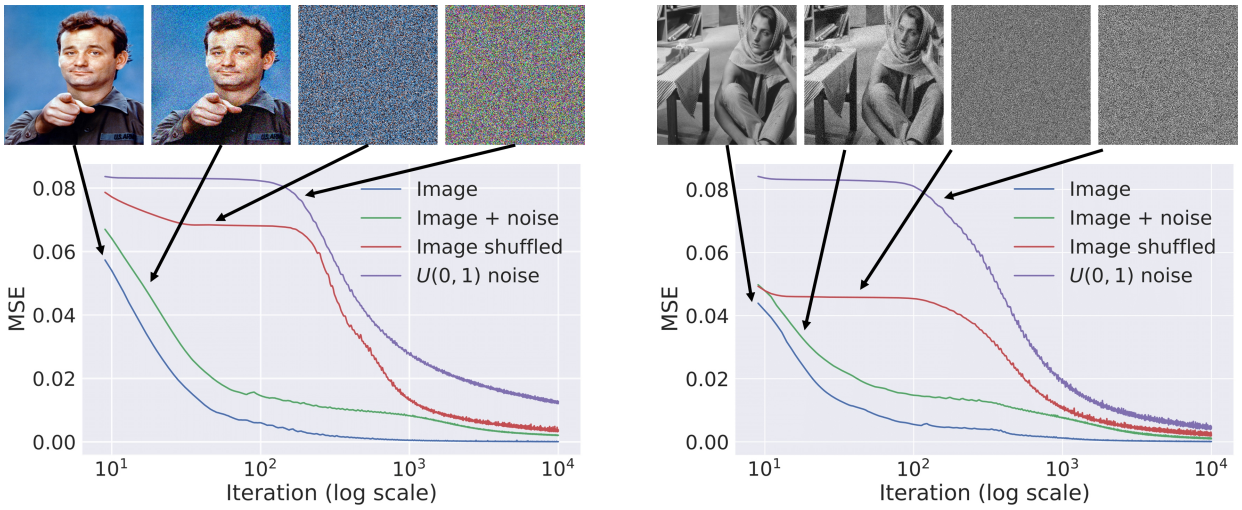


Fig. 4: Learning curves for the reconstruction task using: a natural image, the same plus i.i.d. noise, the same randomly scrambled, and white noise. Naturally-looking images result in much faster convergence, whereas noise is rejected.

the *low-level structure* of natural images. This knowledge is sufficient to model *conditional* image distribution $p(x|x_0)$ where x has to be determined given a corrupted observation x_0. The latter can be used to solve inverse problems such as denoising [10], super-resolution [12], or inpainting.

Rather than working with distributions explicitly, we formulate such tasks 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 is the noisy/low-resolution/occluded image, and $R(x)$ is a regularizer. The choice of data term $E(x; x_0)$ is often directly dictated by the application and is thus not difficult.

The regularizer $R(x)$, on the other hand, is often not tied to a specific application because it captures the generic regularity of natural images. While much research has gone in the design of good regularizers, a simple example is Total Variation (TV), which encourages images to contain uniform regions.

In this work, we drop the explicit regularizer $R(x)$ and use instead the implicit prior captured by the neural network parametrization, as follows:

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

The (local) minimizer θ^* is obtained using an optimizer such as gradient descent, starting from a *random initialization* of the parameters θ (see fig. 2), so as the only information available is the noisy image x_0. Given the resulting (local) minimizer θ^*, the result of the restoration process is obtained as $x^* = f_{\theta^*}(z)$.[1] This approach is schematically depicted in fig. 3 (left).

Since no aspect of the network f_θ is learned from data beforehand, such *deep image prior* is effectively handcrafted, just like the TV norm. The contribution of the paper is to show that this hand-crafted prior works very well for various image restoration tasks, well beyond standard handcrafted priors and approaching learning-based approaches in many cases.

As we show in the experiments, the choice of architecture does have an impact on the results. In particular, most of our experiments are performed using a U-Net-like "hourglass" architecture with skip connections, where z and x have the same spatial dimensions and the network has several millions of parameters. Furthermore, while it is also possible to optimize over the code z, in our experiments we do not do so. Thus, unless noted otherwise, z is a fixed randomly-initialized $3D$ tensor.

2.1 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

[1] Equation (2) can also be thought of as a regularizer $R(x)$ in the style of eq. (1), where $R(x) = 0$ for all images that can be generated by a deep ConvNet of a certain architecture with the weights being not too far from random initialization, and $R(x) = +\infty$ for all other signals.

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 on 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 (fig. 3 – left), or, at least, the optimization trajectory passes near one (fig. 3 – right).

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 4 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 some applications, we restrict the number of iterations in the optimization process (2). 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. The use of deep image prior with the restriction on the number of iterations in the optimization process is schematically depicted in fig. 3 (right).

2.2 "Sampling" from the deep image prior

The prior defined by eq. (2) is implicit and does not define a proper probability distribution in the image space. Nevertheless, it is possible to look at the "samples" (in the loose sense) from this prior, by taking random values of the parameters θ and looking at the generated image $f_\theta(z)$. In other words, we can visualize the starting points of the optimization process eq. (2) before fitting to data starts. Figure 5 shows such "samples" from the deep priors defined by different hourglass-type architectures. The samples exhibit spatial structures and self-similarities, whereas the scale of these structures depends on the depth of the networks. Adding skip connections then results in images that contain structures of different characteristic scales, as is desirable for modeling natural images. It is therefore natural that such architectures are the most popular choice for generative ConvNets. They have also performed best in our image restoration experiments described next.

3 Applications

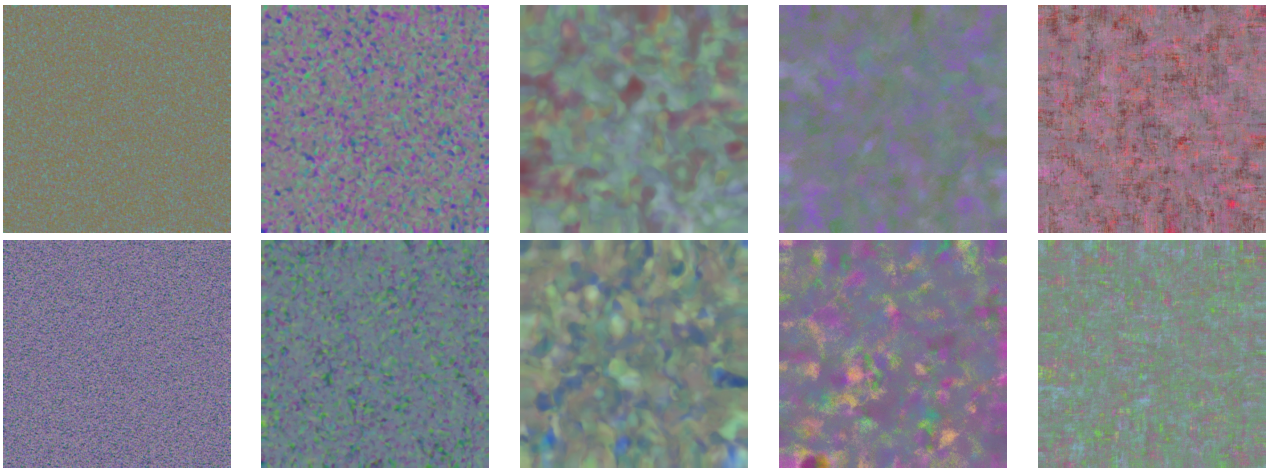
We now show experimentally how the proposed prior works for diverse image reconstruction problems. More examples and interactive viewer can be found on the project webpage https://dmitryulyanov.github.io/deep_image_prior.

3.1 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.

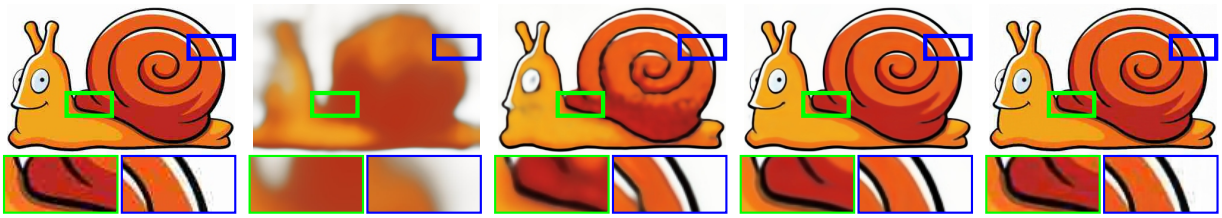
Here we work under the blindness assumption, but the method can be easily modified to incorporate information about noise model. We use the same exact formulation as eqs. (3) and (4) and, given a noisy image x_0, recover a clean image $x^* = f_{\theta^*}(z)$ after substituting the minimizer θ^* of eq. (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 difficult. We demonstrate this capability by several qualitative examples in fig. 7, where our approach uses the quadratic energy (3) leading to formulation (4) to restore images degraded by complex and unknown compression artifacts. Figure 6 (top row) also demonstrates the applicability of the method beyond natural images (a cartoon in this case).



a) Hourglass-1 b) Hourglass-3 c) Hourglass-5 d) Skip-5 e) Skip-5-nearest

Fig. 5: **"Samples" from the deep image prior.** We show images that are produced by ConvNets with random weights from independent random uniform noise. Each column shows two images $f_\theta(z)$ for the same architecture, same input noise z, and two different random θ. The following architectures are visualized: a) an hourglass architecture with one downsampling and one bilinear upsampling, b) a deeper hourglass architecture with three downsamplings and three bilinear upsamplings, c) an even deeper hourglass architecture with five downsamplings and five bilinear upsamplings, d) same as (c), but with skip connections (each skip connection has a convolution layer), e) same as (d), but with nearest upsampling. Note how the resulting images are far from independent noise and correspond to stochastic processes producing spatial structures with clear self-similarity (e.g. each image has a distinctive palette). The scale of structures naturally changes with the depth of the network. "Samples" for hourglass networks with skip connections (U-Net type) combine structures of different scales, as is typical for natural images.



Corrupted 100 iterations 600 iterations 2400 iterations 50K iterations

Fig. 6: **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).

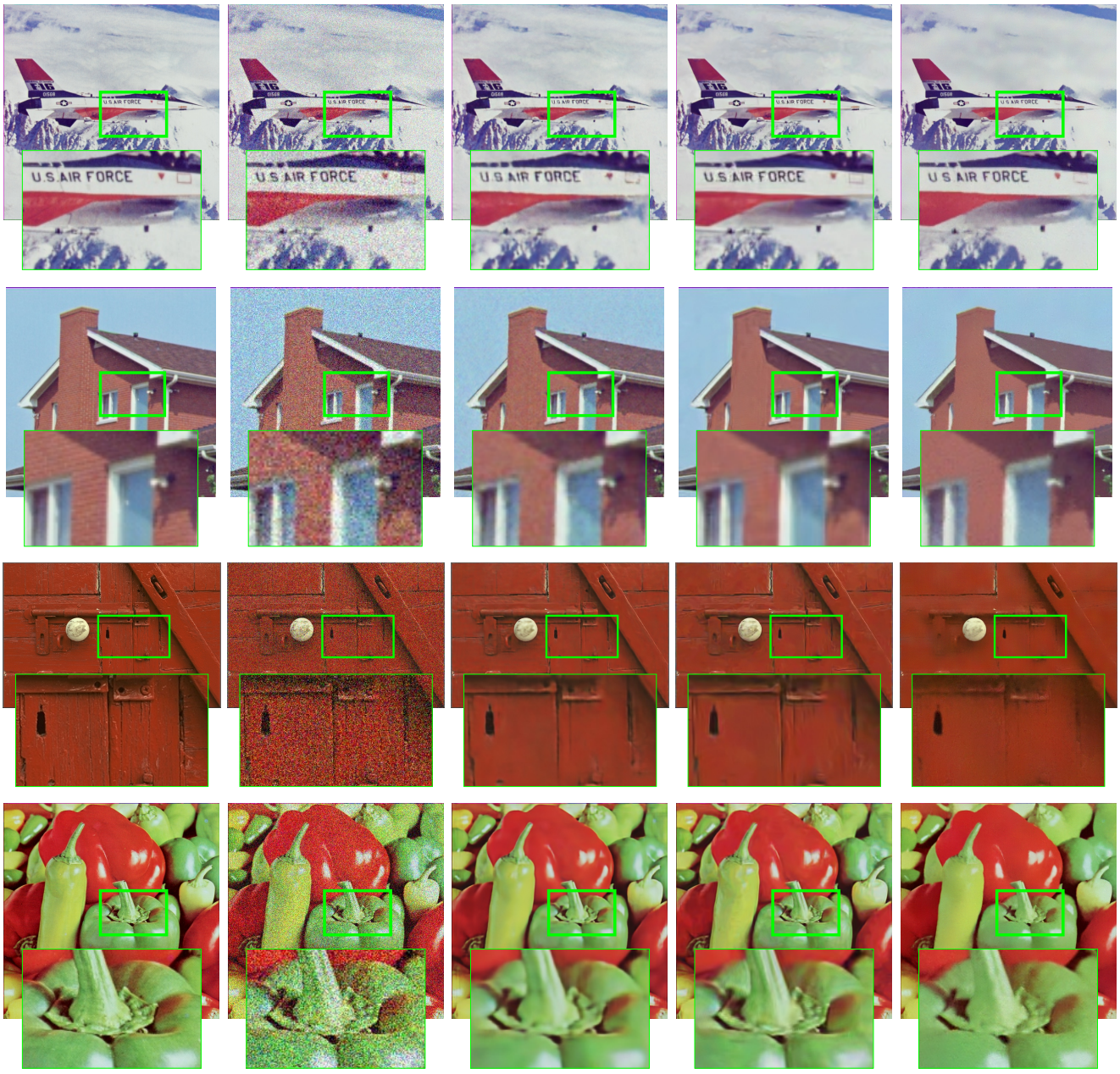
We evaluate our denoising approach on the standard dataset[2], consisting of 9 colored images with noise strength of $\sigma = 25$. We achieve a PSNR of 29.22 after 1800 optimization steps. The score is improved up to 30.43 if we additionally average the restored images obtained in the last iterations (using exponential sliding window). If averaged over two optimization runs our method further improves up to 31.00 PSNR. For reference, the scores for the two popular approaches CMB3D [11] and Non-local means [9], that do not require pretraining, are 31.42 and 30.26 respectively.

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

3.2 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:

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



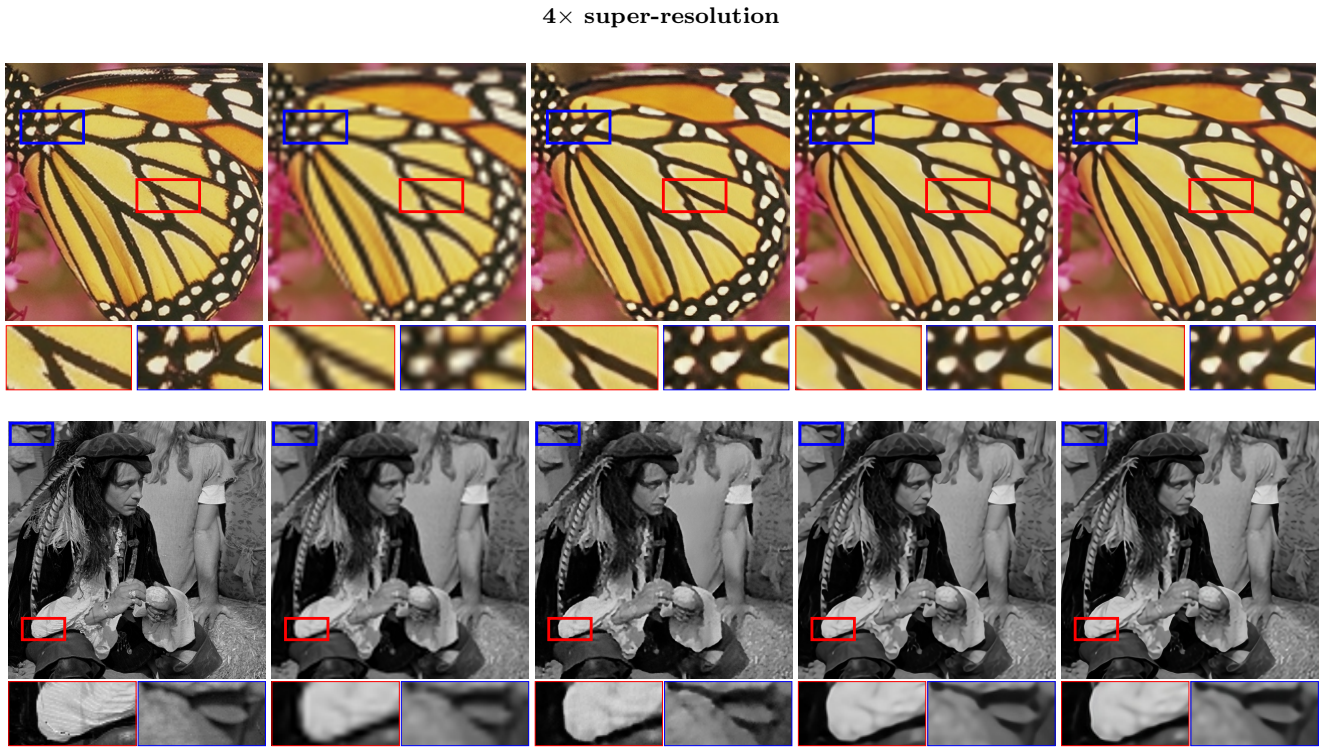
(a) GT (b) Input (c) Ours (d) CBM3D (e) NLM

Fig. 7: **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 [11,9] is shown.

where $d(\cdot) : \mathbb{R}^{3\times tH\times tW} \to \mathbb{R}^{3\times H\times W}$ is a *downsampling operator* that resizes 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 injective). 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 [4] and Set14 [58] datasets. We use a scaling factor of 4 and 8 to compare to other works in fig. 8.



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

(f) Original image (g) Bicubic, **Not trained** (h) Ours, **Not trained** (i) LapSRN, **Trained** (j) VDSR, **Trained**

Fig. 8: **4x and 8x 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 [33], SRResNet [35], VDSR [29]) which utilize networks trained from large datasets.

(a) HR image (b) Bicubic upsampling (c) No prior (d) TV prior (e) Deep image prior

Fig. 9: **Prior effect in super-resolution.** Direct optimization of data term $E(x; x_0)$ with respect to the pixels (c) leads to ringing artifacts. TV prior removes ringing artifacts (d) but introduces cartoon effect. Deep prior (e) leads to the result that is both clean and sharp.

4× super-resolution

	Baboon	Barbara	Bridge	Coastguard	Comic	Face	Flowers	Foreman	Lenna	Man	Monarch	Pepper	Ppt3	Zebra	**Avg.**
No prior	22.24	24.89	23.94	24.62	21.06	29.99	23.75	29.01	28.23	24.84	25.76	28.74	20.26	21.69	24.93
Bicubic	**22.44**	25.15	24.47	25.53	21.59	**31.34**	25.33	29.45	29.84	25.7	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
Glasner et al.	**22.44**	25.38	**24.73**	25.38	21.98	31.09	25.54	30.40	30.48	**26.33**	28.22	32.02	22.16	24.34	26.46
Ours	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**
SRResNet-MSE	23.0	26.08	25.52	26.31	23.44	32.71	28.13	33.8	32.42	27.43	32.85	34.28	26.56	26.95	28.53
LapSRN	22.83	25.69	25.36	26.21	22.9	32.62	27.54	33.59	31.98	27.27	31.62	33.88	25.36	26.98	28.13

8× super-resolution

	Baboon	Barbara	Bridge	Coastguard	Comic	Face	Flowers	Foreman	Lenna	Man	Monarch	Pepper	Ppt3	Zebra	**Avg.**
No prior	21.09	23.04	21.78	23.63	18.65	27.84	21.05	25.62	25.42	22.54	22.91	25.34	18.15	18.85	22.56
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
SelfExSR	21.37	23.90	22.28	24.17	19.79	29.48	**22.93**	27.01	27.72	**23.83**	24.02	28.63	20.09	20.25	23.96
Ours	**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**
LapSRN	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

Table 1: Detailed super-resolution PSNR comparison on the Set14 dataset with different scaling factors.

4× super-resolution

	Baby	Bird	Butterfly	Head	Woman	**Avg.**
No prior	30.16	27.67	19.82	29.98	25.18	26.56
Bicubic	31.78	30.2	22.13	31.34	26.75	28.44
TV prior	31.21	30.43	24.38	31.34	26.93	28.85
Glasner et al.	**32.24**	31.10	22.36	**31.69**	26.85	28.84
Ours	31.49	**31.80**	**26.23**	31.04	**28.93**	**29.89**
LapSRN	33.55	33.76	27.28	32.62	30.72	31.58
SRResNet-MSE	33.66	35.10	28.41	32.73	30.6	32.10

8× super-resolution

	Baby	Bird	Butterfly	Head	Woman	**Avg.**
No prior	26.28	24.03	17.64	27.94	21.37	23.45
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
SelfExSR	**28.45**	26.48	18.80	29.36	24.05	25.42
Ours	28.28	**27.09**	**20.02**	**29.55**	**24.50**	**25.88**
LapSRN	28.88	27.10	19.97	29.76	24.79	26.10

Table 2: Detailed super-resolution PSNR comparison on the Set5 dataset with different scaling factors.

Qualitative comparison with bicubic upsampling and state-of-the art learning-based methods SRResNet [35], LapSRN [52] is presented in fig. 8. 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 learning-based methods that use the MSE loss. GAN-based [19] methods SRGAN [35] and EnhanceNet [48] (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 (tables 1 and 2). While our method is still outperformed by learning-based approaches, it does considerably better than the non-trained ones (bicubic, [18], [26]). Visually, it seems to close most of the gap between non-trained methods and state-of-the-art trained ConvNets (c.f. figs. 1 and 8).

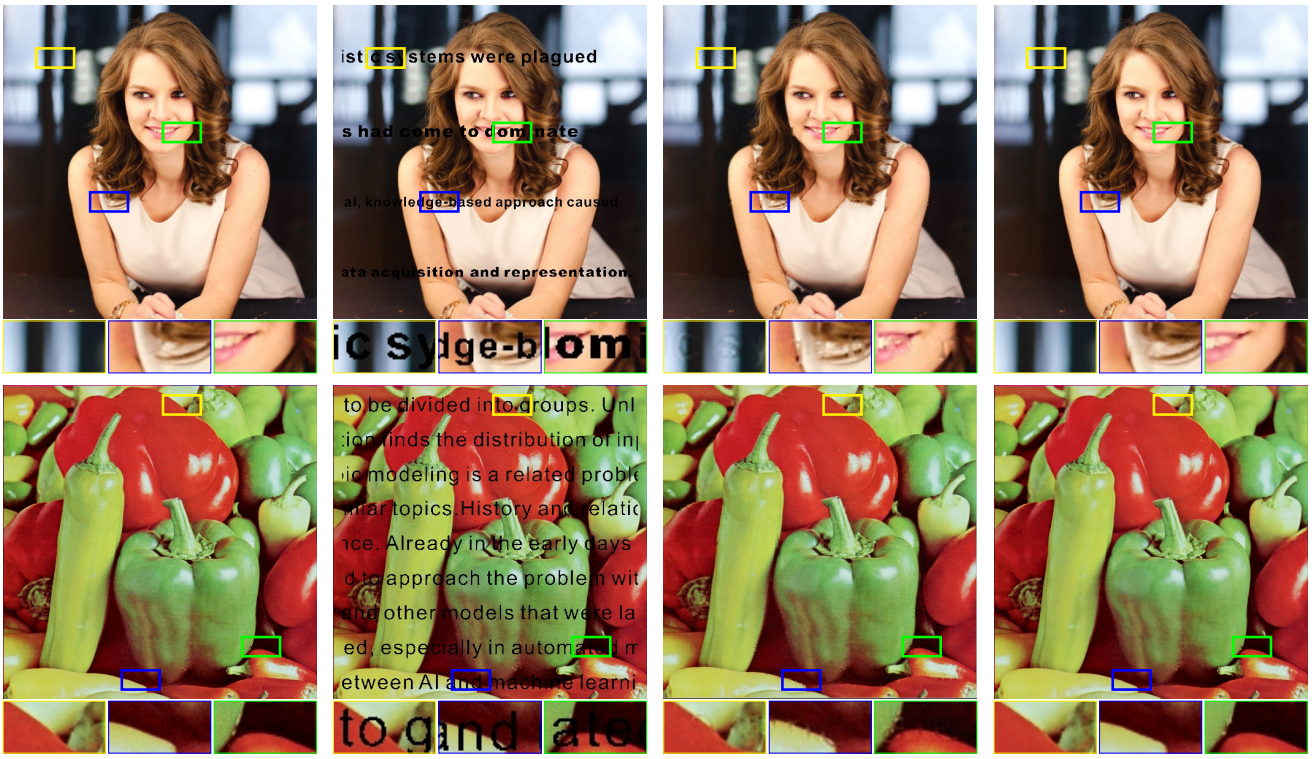
In fig. 9 we compare our deep prior to non-regularized solution and a vanilla TV prior. Our result do not have both ringing artifacts and cartoonish effect.

3.3 Inpainting

In image inpainting, one is given an image x_0 with missing pixels in correspondence of a binary mask $m \in$

(a) Corrupted image (b) Global-Local GAN [27] (c) Ours, LR = 0.01 (d) Ours, LR = 10^{-4}

Fig. 10: **Region inpainting.** In many cases, deep image prior is sufficient to successfully inpaint large regions. Despite using no learning, the results may be comparable to [27] 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 most images we tried.



(a) Original image (b) Corrupted image (c) Shepard networks [44] (d) Deep Image Prior

Fig. 11: Comparison with Shepard networks [44] on text the inpainting task. Even though [44] utilizes learning, the images recovered using our approach look more natural and do not have halo artifacts.

$\{0,1\}^{H\times W}$; the goal is to reconstruct the missing data. The corresponding data term is given by

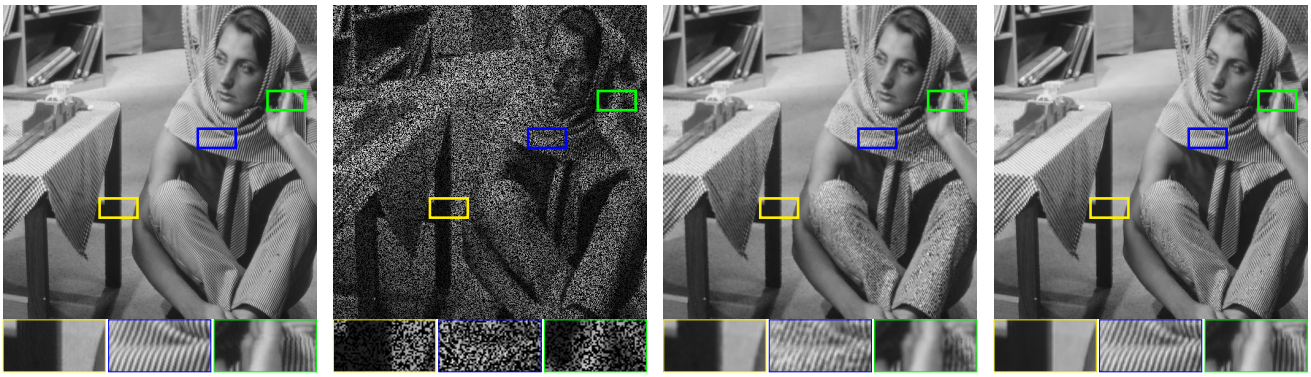
$$E(x;x_0) = \|(x - x_0)\odot m\|^2\,, \tag{6}$$

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. 11) inpainting is used to remove text overlaid on an image. Our approach is compared to the method of [44] specifically designed for inpainting. Our approach leads to an almost perfect results with virtually no artifacts, while for [44] the text mask remains visible in some regions.

	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage
Papyan et al.	28.14	31.44	34.58	35.04	31.11	27.90	31.18	31.34	32.35	31.92	28.05
Ours	**32.22**	**33.06**	**39.16**	**36.16**	**33.05**	**29.8**	**32.52**	**32.84**	**32.77**	**32.20**	**34.54**

Fig. 12: Comparison between our method and the algorithm in [42]. See fig. 13 for visual comparison.



(a) Original image (b) Corrupted image (c) CSC [42] (d) Deep image prior

Fig. 13: Comparison with convolutional sparse coding (CSC) [42] on inpainting 50% of missing pixels. Our approach recovers a natural image with more accurate fine details than convolutional sparse coding.

Next, fig. 13 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 [42] based on convolutional sparse coding. To obtain results for [42] we first decompose the corrupted image x_0 into low and high frequency components similarly to [21] 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 [42]). The quantitative comparison on the standard data set [24] for our method is given in fig. 12, showing a strong quantitative advantage of the proposed approach compared to convolutional sparse coding. In fig. 13 we present a representative qualitative visual comparison with [42].

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 [27] in fig. 10. The deep image prior utilizes context of the image and interpolates the unknown region with textures from the known part. Such behaviour highlights the relation between the deep image prior and traditional self-similarity priors.

In fig. 14, 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.

3.4 Natural pre-image

The natural pre-image method of [37] 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

$$\Phi^{-1}(\Phi(x_0)) = \{x \in \mathcal{X} : \Phi(x) = \Phi(x_0)\} \tag{7}$$

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

$$E(x; x_0) = \|\Phi(x) - \Phi(x_0)\|^2 . \tag{8}$$

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 [37].

In practice, finding points in the natural pre-image can be done by regularizing the data term similarly

(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

Fig. 14: **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.

to the other inverse problems seen above. The authors of [37] prefer to use the TV norm, which is a weak natural image prior, but is relatively unbiased. On the contrary, papers such as [14] learn to invert a neural network from examples, resulting in better looking reconstructions, which however may be biased towards learning 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 [14].

For evaluation, our method is compared to the ones of [38] and [14]. Figure 15 shows the results of inverting representations Φ obtained by considering progres-

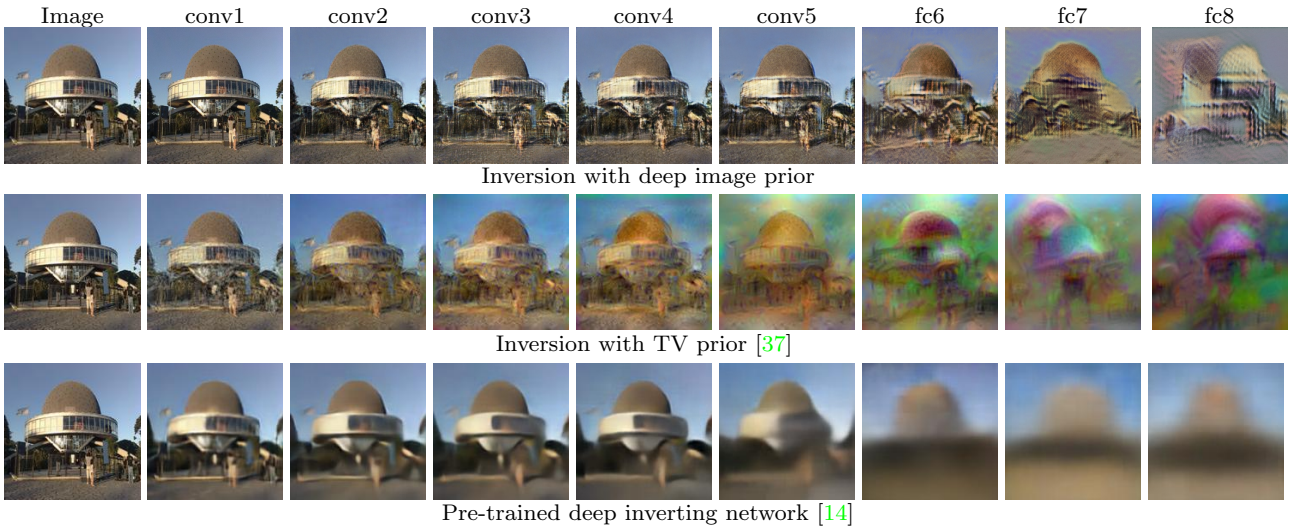


Fig. 15: **AlexNet inversion.** Given the image on the left, we show the natural pre-image obtained by inverting different layers of AlexNet (trained for classification on ImageNet ISLVRC) using three different regularizers: the Deep Image prior, the TV norm prior of [37], and the network trained to invert representations on a hold-out set [14]. The reconstructions obtained with the deep image prior are in many ways at least as natural as [14], yet they are not biased by the learning process.

sively deeper subsets of AlexNet [32]: `conv1`, `conv2`, ..., `conv5`, `fc6`, `fc7`, and `fc8`. Pre-images are found either by optimizing (2) using a structured prior.

As seen in fig. 15, 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 [14], which have a clear tendency to regress to the mean. Note that [14] has been followed-up by [13] where they used a learnable discriminator and a perceptual loss to train the model. While the usage of a more complex loss clearly improved their results we do not compare to their method here.

We perform similar experiment and invert layers of VGG-19 [51] in fig. 16 and also observe an improvement.

3.5 Activation maximization

Along with the pre-image method, the *activation maximization* method is used to visualize internals of a deep neural network. It aims to synthesize an image that highly activates a certain neuron by solving the following optimization problem:

$$x^* = \arg\max_x \Phi(x)_m \,, \tag{9}$$

where m is an index of a chosen neuron. $\Phi(x)_m$ corresponds to m-th output if Φ ends with fully-connected layer and central pixel of the m-th feature map if the $\Phi(x)$ has spatial dimensions.

We compare the proposed deep prior to TV prior from [37] in fig. 17, where we aim to maximize activations of the last `fc8` layer of AlexNet and VGG-16. For AlexNet deep image prior leads to more natural and interpretable images, while the effect is not as clear in the case of VGG-16. In fig. 18 we show more examples, where we maximize the activation for a certain class.

3.6 Image enhancement

We also use the proposed deep image regularization to perform high frequency enhancement in an image. As demonstrated in section 2.1, the noisy image is being reconstructed starting from coarse low-frequency details and finishing with fine high frequency details and noise. To perform enhancement we use the objective 4 setting the target image to be x_0. We stop the optimization process at a certain point, obtaining a coarse approximation x_c of the image x_0. The fine details are then computed as

$$x_f = x_0 - x_c \,. \tag{10}$$

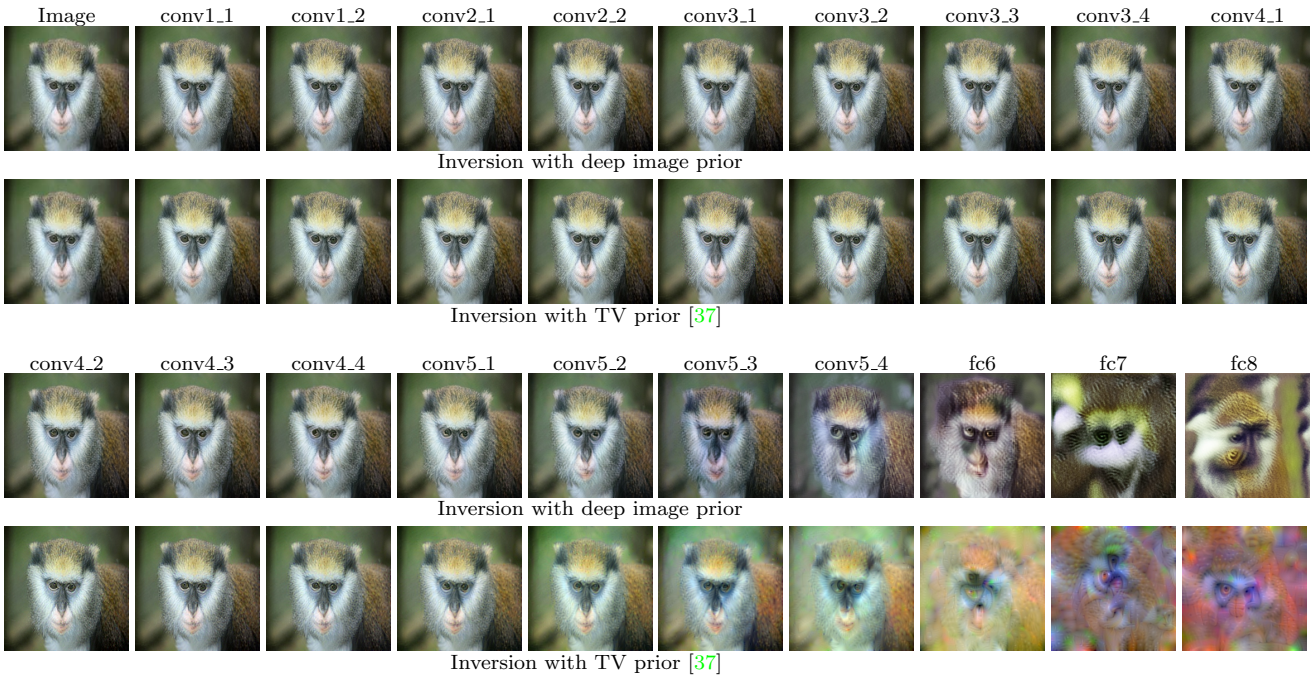


Fig. 16: Inversion of VGG-19 [51] network activations at different layers with different priors.

We then construct an enhanced image by boosting the extracted fine details x_f:

$$x_e = x_0 + x_f \,. \tag{11}$$

In fig. 19 we present coarse and enhanced versions of the same image, running the optimization process for different number of iterations. At the start of the optimization process (corresponds to low number of iteration) the resulted approximation does not precisely recreates the shape of the objects (c.f. blue halo in the bottom row of fig. 19). While the shapes become well-matched with the time, unwanted high frequency details also start to appear. Thus we need to stop the optimization process in time.

3.7 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 [43], 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 [22]. The results of the restoration are given in the fig. 20.

4 Technical details

While other options are possible, we mainly experimented with fully-convolutional architectures, where the input $z \in \mathbb{R}^{C' \times W \times H}$ has the same spatial resolution as the the output of the network $f_\theta(z) \in \mathbb{R}^{3 \times W \times H}$.

We use encoder-decoder ("hourglass") architecture (possibly with skip-connections) for f_θ in all our experiments except noted otherwise (fig. 21), varying small number of hyperparameters. Although the best results can be achieved by carefully tuning an architecture for a particular task (and potentially for a particular image), we found that wide range of hyperparameters and architectures give acceptable results.

We use LeakyReLU [23] as a non-linearity. As a downsampling technique we simply use strides implemented within convolution modules. We also tried aver-

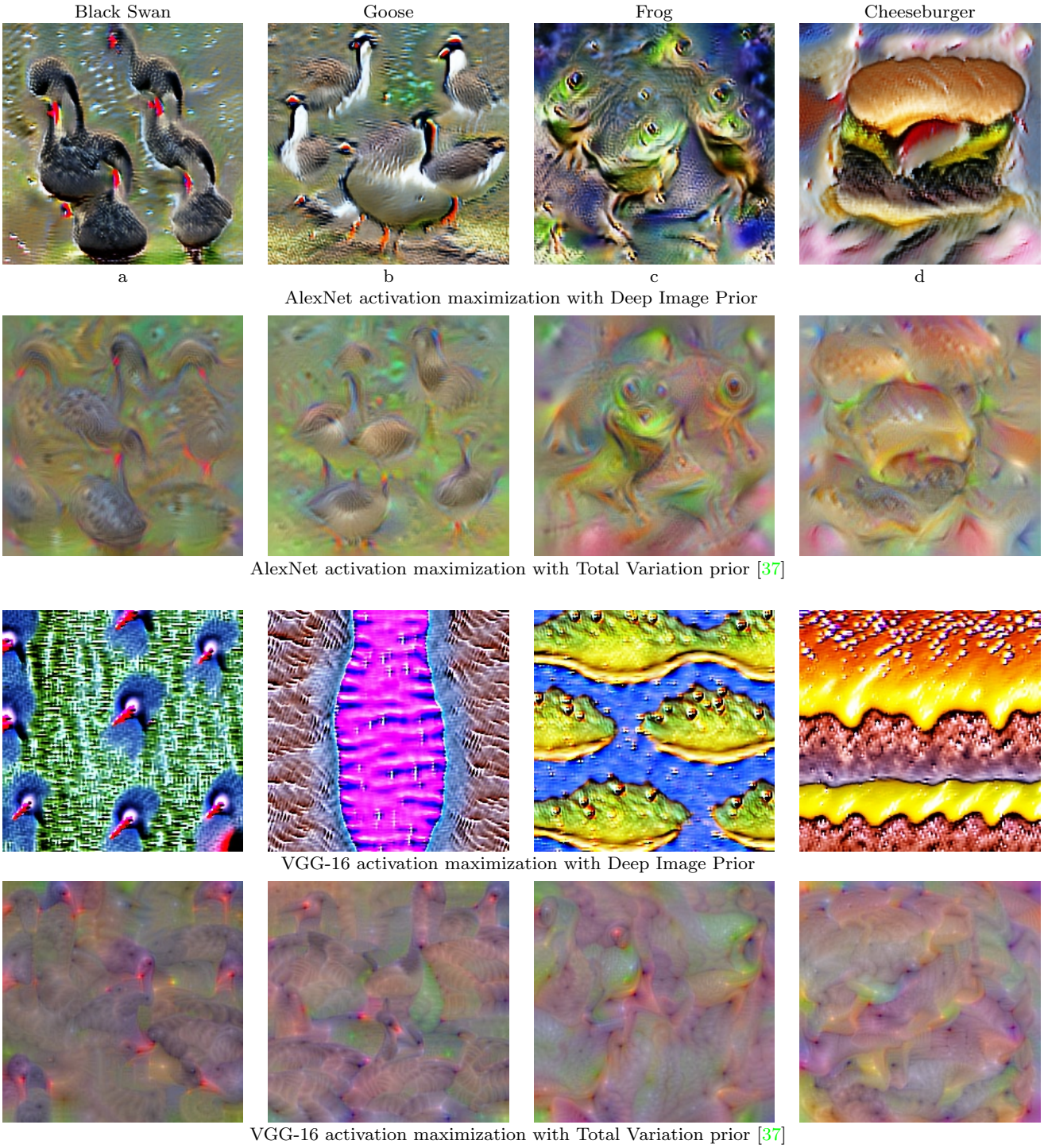


Fig. 17: **Class activation maximization.** For a given class label shown at the very top, we show images obtained by maximizing the corresponding class activation (before soft-max) of AlexNet (top) and VGG-16 (bottom) architectures using different regularizers: the deep image prior proposed here (rows 1 and 3), and the total variation prior of [47]. For both architectures (AlexNet) in particular, inversion with deep image prior leads to more interpretable results.

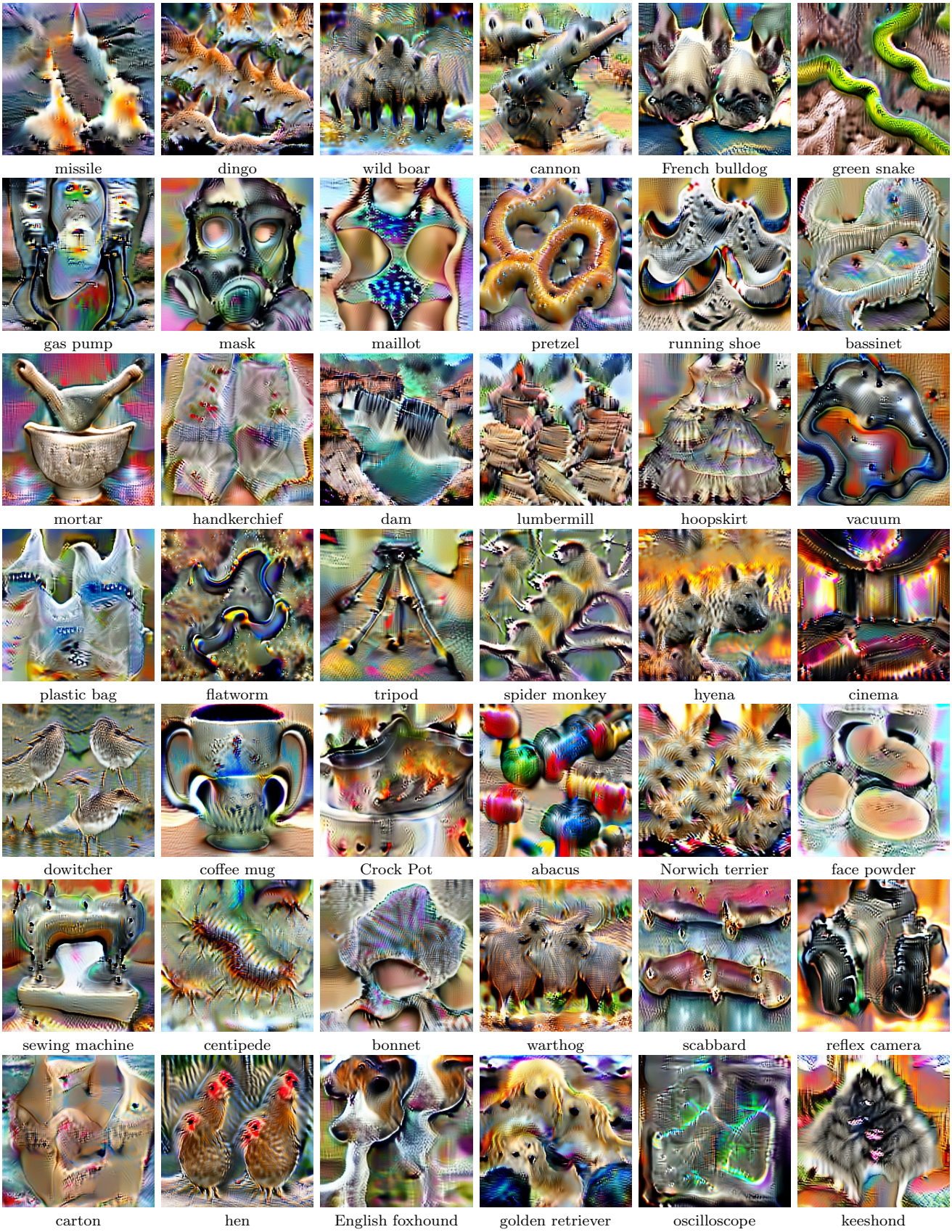
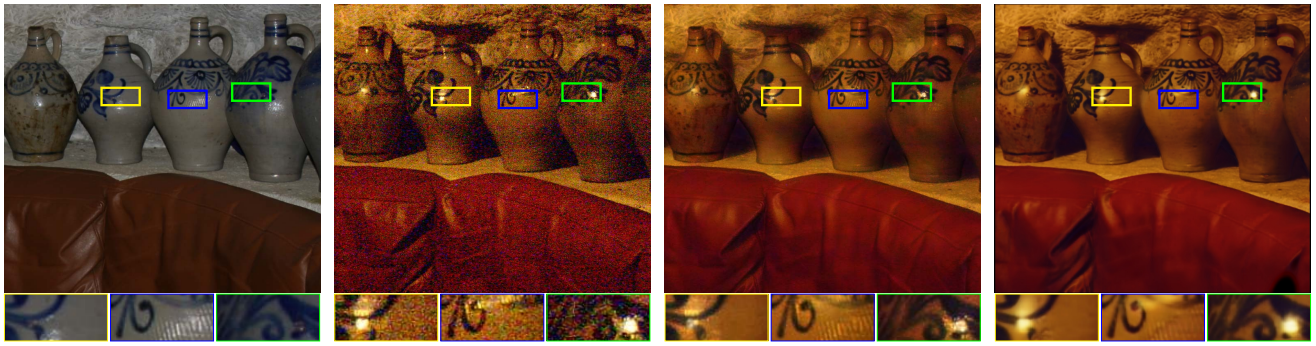


Fig. 18: AlexNet activation maximization regularized with deep image prior for different randomly-selected ILSVRC class labels.

(a) Iteration 250 (b) Iteration 500 (c) Iteration 2500 (d) Iteration 7000

Fig. 19: **Coarse and boosted images for different stopping points.** We obtain the coarse images (second row) running the optimization for reconstruction objective 4 for a certain number of iterations. We then subtract coarse version from the original image to get fine details and boost them (first row). Even for low iteration number the coarse approximation preserves edges for the large objects. The original image in shown the first column.



(a) Flash (b) No flash (c) Joint bilateral [43] (d) Deep image prior

Fig. 20: **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 [43] (c.f. blue inset).

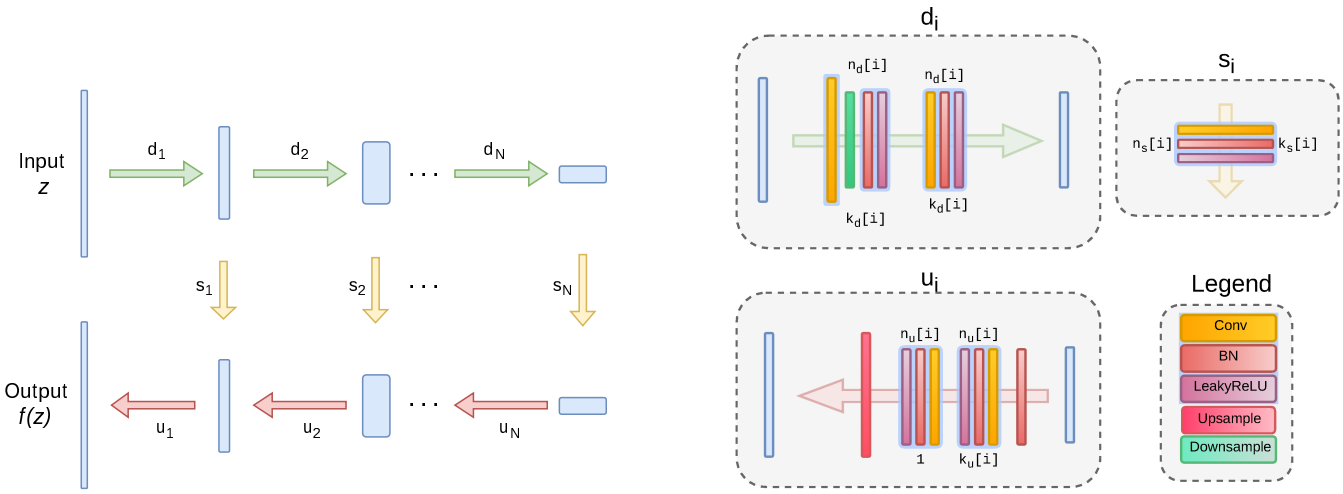


Fig. 21: **The architecture used in the experiments.** We use "hourglass" (also known as "decoder-encoder") architecture. We sometimes add skip connections (yellow arrows). $n_u[i]$, $n_d[i]$, $n_s[i]$ correspond to the number of filters at depth i for the upsampling, downsampling and skip-connections respectively. The values $k_u[i]$, $k_d[i]$, $k_s[i]$ correspond to the respective kernel sizes.

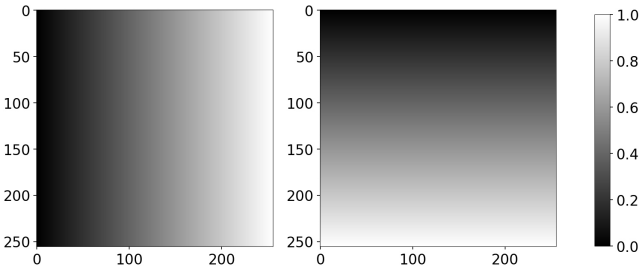


Fig. 22: **"Meshgrid" input** z used in some inpainting experiments. These are two channels of the input tensor; in BCHW layout: `z[0, 0, :, :]`, `z[0, 1, :, :]` The intensity encodes the value: from zero (black) to one (white). Such type of input can be regarded as a part of the prior which enforces smoothness.

age/max pooling and downsampling with Lanczos kernel, but did not find a consistent difference between any of them. As an upsampling operation we choose between bilinear upsampling and nearest neighbour upsampling. An alternative upsampling method could be to use transposed convolutions, but the results we obtained using them were worse. We use reflection padding instead of zero padding in convolution layers everywhere except for the feature inversion and activation maximization experiments.

We considered two ways to create the input z: 1. *random*, where the z is filled with uniform noise between zero and 0.1, 2. *meshgrid*, where we initialize $z \in \mathbb{R}^{2\times W\times H}$ using `np.meshgrid` (see fig. 22). Such initialization serves as an additional smoothness prior to the one imposed by the structure of f_θ itself. We found such input to be beneficial for large-hole inpainting, but not for other tasks.

During fitting of the networks we often use a *noise-based regularization*. I.e. at each iteration we perturb the input z with an additive normal noise with zero mean and standard deviation σ_p. While we have found such regularization to impede optimization process, we also observed that the network was able to eventually optimize its objective to zero no matter the variance of the additive noise (i.e. the network was always able to adapt to any reasonable variance for sufficiently large number of optimization steps).

We found the optimization process tends to destabilize as the loss goes down and approaches a certain value. Destabilization is observed as a significant loss increase and blur in generated image $f_\theta(z)$. From such destabilization point the loss goes down again till destabilized one more time. To remedy this issue we simply track the optimization loss and return to parameters from the previous iteration if the loss difference between two consecutive iterations is higher than a certain threshold.

Finally, we use *ADAM* optimizer [30] in all our experiments and PyTorch as a framework.

Below, we provide the remaining details of the network architectures. We use the notation introduced in fig. 21.

Super-resolution (*default architecture*).

$z \in \mathbb{R}^{32 \times W \times H} \sim U(0, \frac{1}{10})$
$n_u = n_d =$ `[128, 128, 128, 128, 128]`
$k_u = k_d =$ `[3, 3, 3, 3, 3]`
$n_s =$ `[4, 4, 4, 4, 4]`
$k_s =$ `[1, 1, 1, 1, 1]`
$\sigma_p = \frac{1}{30}$
`num_iter` $= 2000$
`LR` $= 0.01$
`upsampling` $=$ `bilinear`

The decimation operator d is composed of low pass filtering operation using Lanczos2 kernel (see [54]) and resampling, all implemented as a single (fixed) convolutional layer.

For `8x` super-resolution (fig. 8) we have changed the standard deviation of the input noise to $\sigma_p = \frac{1}{20}$ and the number of iterations to 4000.

Text inpainting (fig. 11). We used the same hyperparameters as for super-resolution but optimized the objective for 6000 iterations.

Large hole inpainting (fig. 10). We used the same hyperparameters as for super-resolution, but used `meshgrid` as an input, removed skip connections and optimized for 5000 iterations.

Large hole inpainting (fig. 14). We used the following hyperparameters:

$z \in \mathbb{R}^{32 \times W \times H} \sim U(0, \frac{1}{10})$
$n_u = n_d =$ `[16, 32, 64, 128, 128, 128]`
$k_d =$ `[3, 3, 3, 3, 3, 3]`
$k_u =$ `[5, 5, 5, 5, 5, 5]`
$n_s =$ `[0, 0, 0, 0, 0, 0]`
$k_s =$ `[NA, NA, NA, NA, NA, NA]`
$\sigma_p = 0$
`num_iter = 5000`
`LR = 0.1`
`upsampling` $=$ `nearest`

In figures fig. 14 (c), (d) we simply sliced off last layers to get smaller depth.

Denoising (fig. 7). Hyperparameters were set to be the same as in the case of super-resolution with only difference in iteration number, which was set to 1800. We used the following implementations of referenced denoising methods: [34] for CBM3D and [8] for NLM. We used exponential sliding window with weight $\gamma = 0.99$.

JPEG artifacts removal (fig. 6). Although we could use the same setup as in other denoising experiments, the hyperparameters we used to generate the image in fig. 6 were the following:

$z \in \mathbb{R}^{3 \times W \times H} \sim U(0, \frac{1}{10})$
$n_u = n_d =$ `[8, 16, 32, 64, 128]`
$k_u = k_d =$ `[3, 3, 3, 3, 3]`
$n_s =$ `[0, 0, 0, 4, 4]`
$k_s =$ `[NA, NA, NA, 1, 1]`
$\sigma_p = \frac{1}{30}$
`num_iter = 2400`
`LR = 0.01`
`upsampling` $=$ `bilinear`

Image reconstruction (fig. 13). We used the same setup as in the case of superresolution and denoising, but set `num_iter = 11000`, `LR = 0.001`.

Natural pre-image (figs. 15 and 16).

$z \in \mathbb{R}^{32 \times W \times H} \sim U(0, \frac{1}{10})$
$n_u = n_d =$ `[16, 32, 64, 128, 128, 128]`
$k_u = k_d =$ `[7, 7, 5, 5, 3, 3]`
$n_s =$ `[4, 4, 4, 4, 4]`
$k_s =$ `[1, 1, 1, 1, 1]`
`num_iter = 3100`
`LR = 0.001`
`upsampling` $=$ `nearest`

We used `num_iter = 10000` for the VGG inversion experiment (fig. 16)

Activation maximization (figs. 17 and 18). In this experiment we used a very similar set of hyperparameters to the ones in pre-image experiment.

$z \in \mathbb{R}^{32 \times W \times H} \sim U(0, \frac{1}{10})$
$n_u = n_d =$ `[16, 32, 64, 128, 128, 128]`
$k_u = k_d =$ `[5, 3, 5, 5, 3, 5]`
$n_s =$ `[0, 4, 4, 4, 4]`
$k_s =$ `[1, 1, 1, 1, 1]`
`num_iter = 3100`
`LR = 0.001`
`upsampling` $=$ `bilinear`
$\sigma_p = 0.03$

Image enchancement (fig. 19) We used the same setup as in the case of superresolution and denoising, but set $\sigma_p = 0$.

5 Related work

Our approach is obviously related to image restoration and synthesis methods based on learnable ConvNets and referenced above. Here, we review other lines of work related to our approach.

Modelling "translationally-invariant" statistics of natural images using filter responses has a very long history of research. The statistics of responses to various non-random filters (such as simple operators and higher-order wavelets) have been studied in seminal works [17,

40,50,60]. Later, [25] noted that image response distribution w.r.t. random unlearned filters have very similar properties to the distributions of wavelet filter responces.

Our approach is strongly related to a group of restoration methods that avoid training on the hold-out set and exploit the well-studied self-similarity properties of natural images [46,53]. This group includes methods based on joint modeling of groups of similar patches inside corrupted image [9,11,18], which are particularly useful when the corruption process is complex and highly variable (e.g. spatially-varying blur [3]).

In this group, an interesting parallel work with clear links to our approach is the zero-shot super-resolution approach [1], which trains a feed-forward superresolution ConvNet based on synthetic dataset generated from the patches of a single image. While clearly related, the approach [1] is somewhat complementary as it exploit self-similarities across multiple scales of the same image, while our approach exploits self-similarities within the same scale (at multiple scales).

Several lines of work use dataset-based learning and modeling images using convolutional operations. Learning priors for natural images that facilitate restoration by enforcing filter responses for certain (learned) filters is behind an influential field-of-experts model [45]. Also in this group are methods based on fitting dictionaries to the patches of the corrupted image [39,58] as well as methods based on convolutional sparse coding [20, 7]. The connections between convolutional sparse coding and ConvNets are investigated in [41] in the context of recognition tasks. More recently in [42], a fast single-layer convolutional sparse coding is proposed for reconstruction tasks. The comparison of our approach with [42] (figs. 11 and 12) however suggests that using deep ConvNet architectures popular in modern deep learning-based approaches may lead to more accurate restoration results.

Deeper convolutional models of natural images trained on large datasets have also been studied extensively. E.g. deconvolutional networks [57] are trained by fitting hierarchies of representations linked by convolutional operators to datasets of natural images. The recent work [36] investigates the model that combines ConvNet with a self-similarity based denoising and thus bridges learning on image datasets and exploiting within-image self-similarities.

Finally, we note that this manuscript expands the conference version [55] in multiple ways: 1) It gives more intuition, provides more visualizations and explanation for the presented method altogether with extensive technical details. 2) It contains a more thorough experimental evaluation and shows an application to activation maximization and high frequency enchancement. Since the publication of the preliminary version of our approach, it has also been used by other groups in different ways. Thus, [56] proposes a novel method for compressed sensing recovery using deep image prior. The work [2] learns a latent variable model, where the latent space is parametrized by a convolutional neural network. The approach [49] aims to reconstruct an image from an event-based camera and utilizes deep image prior framework to estimate sensor's ego-motion. The method [28] succesively applies deep image prior to defend against adversarial attacks. Deep image prior is also used in [6] to perform phase retrieval for Fourier ptychography.

6 Discussion

We have investigated the success of recent image generator neural networks, teasing apart the contribution of the prior imposed by the choice of architecture from the contribution of the information transferred from external images through learning. In particular, we have shown that fitting a randomly-initialized ConvNet to corrupted images works as a "Swiss knife" for restoration problems. This approach is probably too slow to be useful for most practical applications, and for each particular application, a feedforward network trained for that particular application would do a better job and do so much faster. However, the success of our approach across a wide variety of tasks suggests that an implicit prior inside deep convolutional network architectures is well-suited for image restoration tasks.

Why does this prior emerge, and, more importantly, why does it fit the structure of natural images so well? We speculate that generation by convolutional operations naturally tends impose self-similarity of the generated images (c.f. fig. 5), as convolutional filters are applied across the entire visual field thus imposing certain stationarity on the output of convolutional layers. Hourglass architectures with skip connections naturally impose self-similarity at multiple scales, making the corresponding priors suitable for the restoration of natural images.

We note that our results go largely against the common narrative that explain the success of deep learning in image restoration (and beyond) by the ability to learn rather than by hand-craft priors; instead, we show that properly *hand-crafted* network architectures correspond to better *hand-crafted* priors, and it seems that learning ConvNets builds on this basis. This observation also validates the importance of developing new deep learning architectures.

6.1 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.

References

1. Assaf Shocher Nadav Cohen, M.I.: "zero-shot" super-resolution using deep internal learning. In: CVPR. IEEE Computer Society (2018) 20
2. Athar, S., Burnaev, E., Lempitsky, V.S.: Latent convolutional models. CoRR (2018) 20
3. Bahat, Y., Efrat, N., Irani, M.: Non-uniform blind deblurring by reblurring. In: Proc. CVPR, pp. 3286–3294. IEEE Computer Society (2017) 20
4. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC, pp. 1–10 (2012) 7
5. Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks. CoRR (2017) 2
6. Boominathan, L., Maniparambil, M., Gupta, H., Baburajan, R., Mitra, K.: Phase retrieval for fourier ptychography under varying amount of measurements. CoRR (2018) 20
7. Bristow, H., Eriksson, A.P., Lucey, S.: Fast convolutional sparse coding. In: CVPR, pp. 391–398. IEEE Computer Society (2013) 20
8. Buades, A.: NLM demo. http://demo.ipol.im/demo/bcm_non_local_means_denoising/ 19
9. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. CVPR, vol. 2, pp. 60–65. IEEE Computer Society (2005) 6, 7, 20
10. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: CVPR, pp. 2392–2399 (2012) 2, 4
11. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8), 2080–2095 (2007) 6, 7, 20
12. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Proc. ECCV, pp. 184–199 (2014) 4
13. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: NIPS, pp. 658–666 (2016) 13
14. Dosovitskiy, A., Brox, T.: Inverting convolutional networks with convolutional networks. In: CVPR. IEEE Computer Society (2016) 2, 12, 13
15. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: Proc. CVPR, pp. 1538–1546 (2015) 2
16. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. Tech. Rep. Technical Report 1341, University of Montreal (2009) 2
17. Field, D.J.: Relations between the statistics of natural images and the response properties of cortical cells. Josa a **4**(12), 2379–2394 (1987) 20
18. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proc. ICCV, pp. 349–356 (2009) 9, 20
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. NIPS, pp. 2672–2680 (2014) 2, 9
20. Grosse, R.B., Raina, R., Kwong, H., Ng, A.Y.: Shift-invariance sparse coding for audio classification. In: UAI, pp. 149–158. AUAI Press (2007) 20
21. Gu, S., Zuo, W., Xie, Q., Meng, D., Feng, X., Zhang, L.: Convolutional sparse coding for image super-resolution. In: ICCV, pp. 1823–1831. IEEE Computer Society (2015) 11
22. He, K., Sun, J., Tang, X.: Guided image filtering. T-PAMI **35**(6), 1397–1409 (2013) 14
23. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: CVPR, pp. 1026–1034. IEEE Computer Society (2015) 14
24. Heide, F., Heidrich, W., Wetzstein, G.: Fast and flexible convolutional sparse coding. In: CVPR, pp. 5135–5143. IEEE Computer Society (2015) 11
25. Huang, J., Mumford, D.: Statistics of natural images and models. In: CVPR, pp. 1541–1547. IEEE Computer Society (1999) 20
26. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR, pp. 5197–5206. IEEE Computer Society (2015) 9
27. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and Locally Consistent Image Completion. ACM Transactions on Graphics (Proc. of SIGGRAPH) **36**(4), 107:1–107:14 (2017) 10, 11
28. Ilyas, A., Jalal, A., Asteri, E., Daskalakis, C., Dimakis, A.G.: The robust manifold defense: Adversarial training using generative models. CoRR (2017) 20
29. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR, pp. 1646–1654. IEEE Computer Society (2016) 8
30. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR (2014) 18
31. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proc. ICLR (2014) 2
32. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012) 13
33. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. IEEE Computer Society (2017) 2, 8
34. Lebrun, M.: BM3D code. https://github.com/gfacciol/bm3d 19
35. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. IEEE Computer Society (2017) 1, 2, 8, 9
36. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: CVPR. IEEE Computer Society (2016) 2, 20
37. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: CVPR. IEEE Computer Society (2015) 2, 11, 12, 13, 14, 15
38. Mahendran, A., Vedaldi, A.: Visualizing deep convolutional neural networks using natural pre-images. IJCV (2016) 12

39. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research **11**(Jan), 19–60 (2010) 20
40. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. PAMI **11**(7), 674–693 (1989) 20
41. Papyan, V., Romano, Y., Elad, M.: Convolutional neural networks analyzed via convolutional sparse coding. Journal of Machine Learning Research **18**(83), 1–52 (2017) 20
42. Papyan, V., Romano, Y., Sulam, J., Elad, M.: Convolutional dictionary learning via local processing. In: ICCV. IEEE Computer Society (2017) 11, 20
43. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M.F., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM Trans. Graph. **23**(3), 664–672 (2004) 14, 17
44. Ren, J.S.J., Xu, L., Yan, Q., Sun, W.: Shepard convolutional neural networks. In: NIPS, pp. 901–909 (2015) 10
45. Roth, S., Black, M.J.: Fields of experts. CVPR **82**(2), 205–229 (2009) 20
46. Ruderman, D.L., Bialek, W.: Statistics of natural images: Scaling in the woods. In: NIPS, pp. 551–558. Morgan Kaufmann (1993) 20
47. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. In: Proceedings of the Eleventh Annual International Conference of the Center for Nonlinear Studies on Experimental Mathematics : Computational Issues in Nonlinear Science: Computational Issues in Nonlinear Science, pp. 259–268. Elsevier North-Holland, Inc., New York, NY, USA (1992) 15
48. Sajjadi, M.S.M., Scholkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: The IEEE International Conference on Computer Vision (ICCV) (2017) 9
49. Shedligeri, P.A., Shah, K., Kumar, D., Mitra, K.: Photorealistic image reconstruction from hybrid intensity and event based sensor. CoRR (2018) 20
50. Simoncelli, E.P., Adelson, E.H.: Noise removal via bayesian wavelet coring. In: ICIP (1), pp. 379–382. IEEE Computer Society (1996) 20
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR (2014) 13, 14
52. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. IEEE Computer Society (2017) 2, 9
53. Turiel, A., Mato, G., Parga, N., Nadal, J.: Self-similarity properties of natural images. In: NIPS, pp. 836–842. The MIT Press (1997) 20
54. Turkowski, K.: Filters for common resampling-tasks. Graphics gems pp. 147–165 (1990) 19
55. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: CVPR. IEEE Computer Society (2018) 20
56. Veen, D.V., Jalal, A., Price, E., Vishwanath, S., Dimakis, A.G.: Compressed sensing with deep image prior and learned regularization. CoRR (2018) 20
57. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Proc. CVPR, pp. 2528–2535. IEEE Computer Society (2010) 20
58. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces, *Lecture Notes in Computer Science*, vol. 6920, pp. 711–730. Springer (2010) 7, 20
59. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: ICLR (2017) 2
60. Zhu, S.C., Mumford, D.: Prior learning and gibbs reaction-diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **19**(11), 1236–1250 (1997) 20