Deep Image Prior

Dmitry Ulyanov Andrea Vedaldi Victor Lempitsky

Presented by Maxim Ryabinin

April 27, 2018

Talk outline

Image restoration task

- Definition
- Formulation as optimization problem

Deep Image Prior

- Output image parametrization
- Network structure as a prior
- High noise impedance
- Algorithm step by step

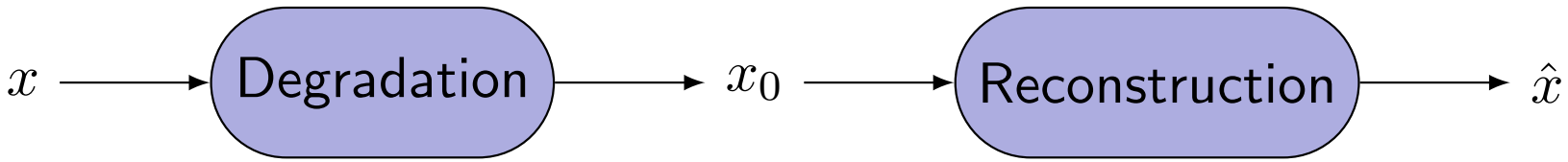
Experiments

- Main architecture
- Tasks solved
- Effect of prior
- Architecture choice

Image restoration

- x — source image
- x_0 — corrupted image (observed)
- $\hat{x}$ — restored image

Given the corrupted image, find the original one



Restoration as optimization

We can express image reconstruction as the following problem:

$$x^* = \arg\min_x \underbrace{E(x; x_0)}_{\text{data term}} + \underbrace{R(x)}_{\text{regularizer}}$$

- E is **task-dependent** and keeps result close to x_0
  Example: MSE for denoising

$$E(x; x_0) := \|x - x_0\|^2$$

- R contains **prior information** about images in general
  Usually handcrafted or trained
  Example: TV (total variation) norm

$$R(x) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{1}{2}}$$

Output image parametrization

What if we generate our image from noise by a parametric function (e.g. a neural network) and optimize wrt its weights?

$$\textbf{Regular: } x^* = \arg\min_x E(x; x_0) + R(x)$$

$$\textbf{Parametrized: } \theta^* = \arg\min_\theta E(f_\theta(z); x_0) + R(f_\theta(z))$$

- Initial parameters are **random**, z is a (fixed) noise tensor
- We are now searching for the best solution in the space of network parameters instead of image space
- Best reconstruction x^* is obtained as $f_{\theta^*}(z)$
- Need heavy regularization... or not?

Network structure as a prior

- Not all images are easy to obtain by every architcture. What if we use this as a regularizer?

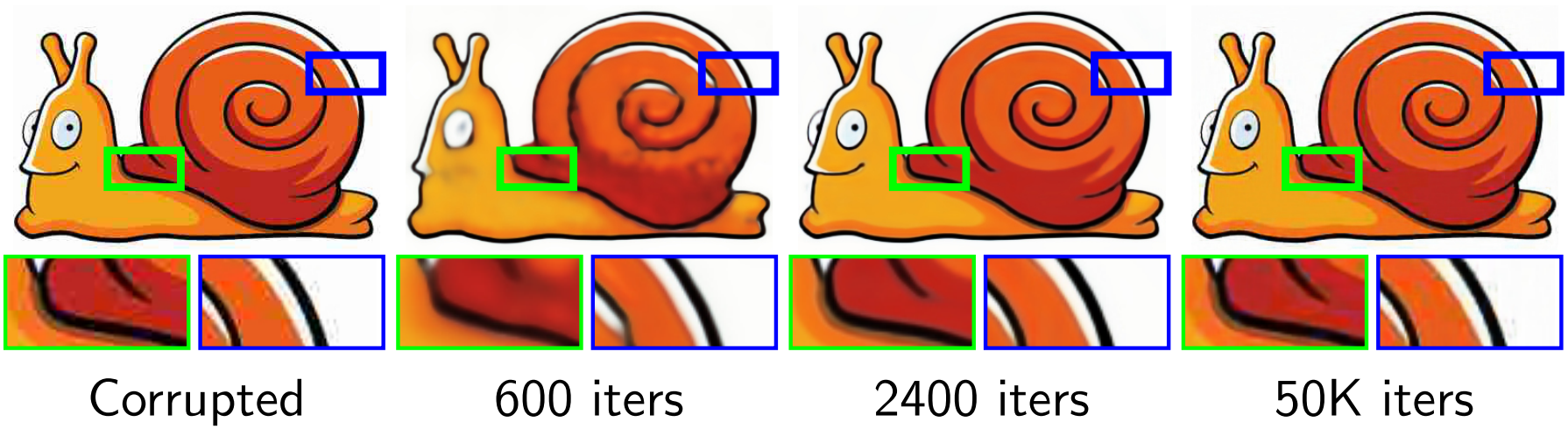
$$R(x) = \begin{cases} 0, \exists \theta : f_\theta(z) = x, \\ +\infty, \text{ otherwise} \end{cases}$$

- This means we replace the regularizer by implicit prior captured by network structure (recall that it is not trained)
- Now we optimize only the data term

$$\theta^* = \arg\min_\theta E(f_\theta(z); x_0)$$

Does this actually work?

- We know that deep networks easily overfit
- Why won't we simply recover the input?
- In the limit — yes. Let's fix number of iterations to avoid that
- This corresponds to reducing set of parameters to those that are "not too far" from initial θ_0



Blind restoration of a JPEG-compressed image

High noise impedance

The network resists "bad" solutions and descends much more quicker towards naturally-looking images

Image

Image + noise

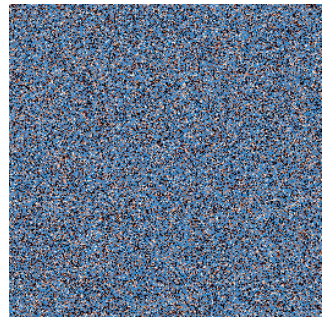
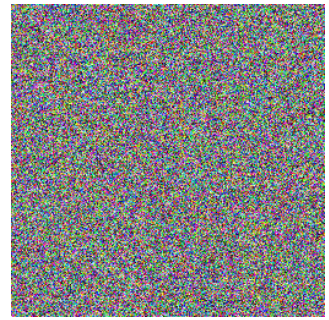
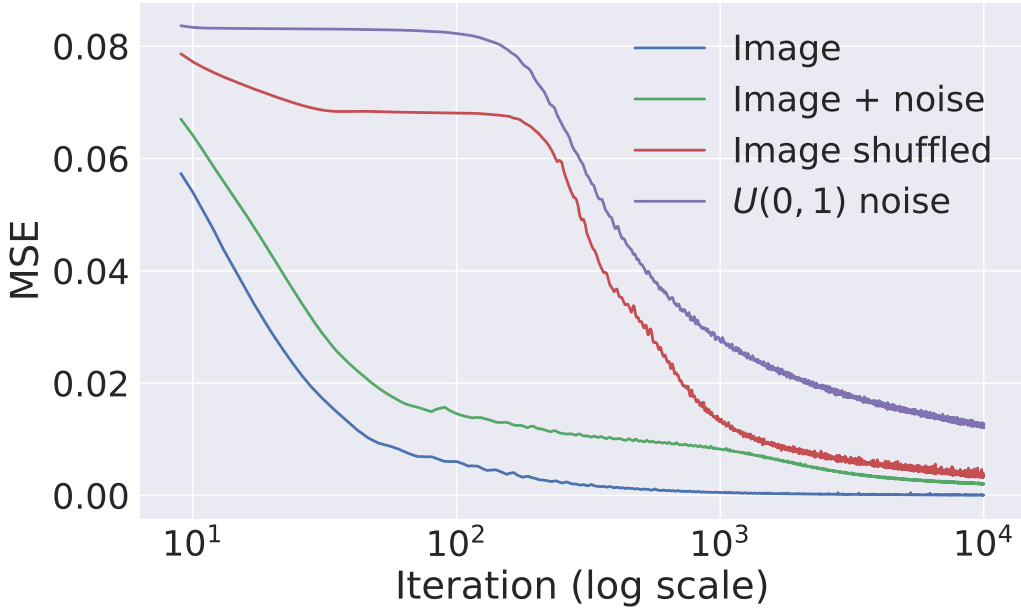


Image shuffled



$U(0,1)$ noise



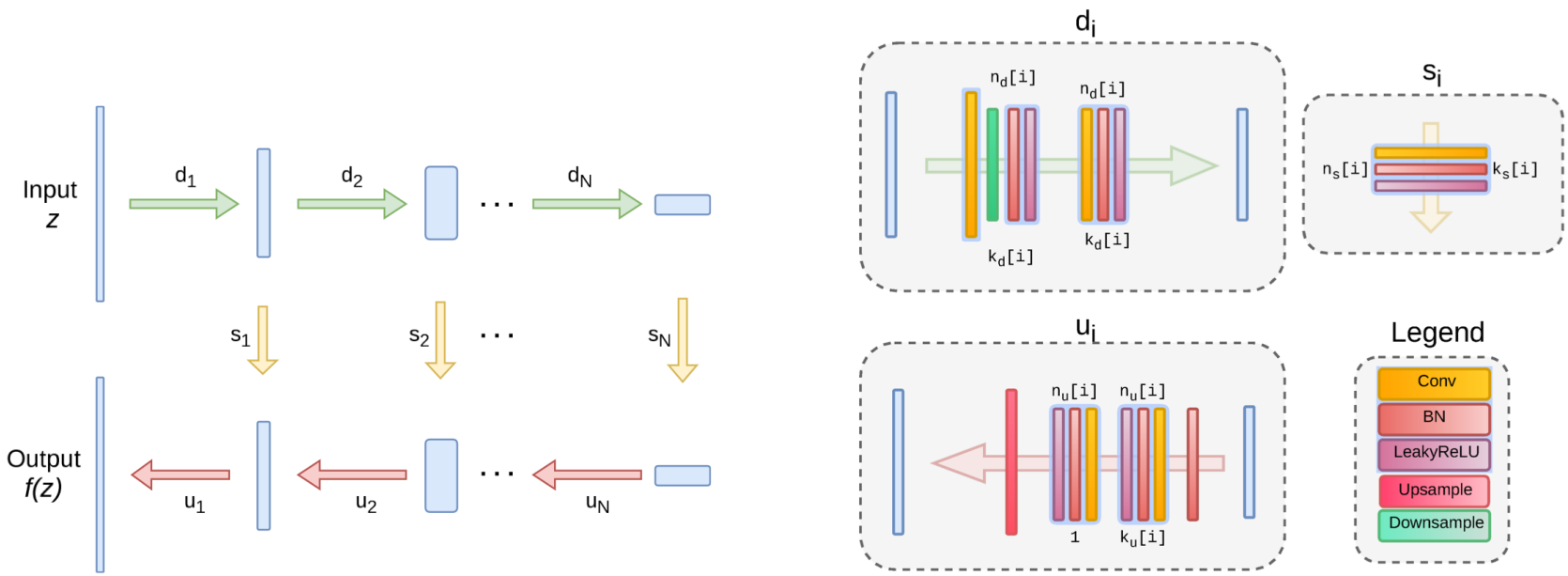
Learning curves for the reconstruction task

Deep Image Prior step by step

1. x_0 — corrupted image (observed)
2. Initialize z (for example, fill it with uniform noise)
3. Solve
$$\theta^* = \arg\min_{\theta} E(f_\theta(z); x_0)$$
with your favorite GD modification (authors used Adam) for `num_iter` iterations
4. Get the result
$$\hat{x} = x^* = f_{\theta^*}(z)$$

Main architecture

- n_u,n_d,n_s — number of convolution filters
- k_u,k_d,k_s — kernel sizes
- Downsample is implemented as strided convolution
- Upsampling is either bilinear or nearest neighbor
- Sometimes skip connections are used



Architecture used in the experiments

Tasks considered

As was mentioned before, data term is specific to each task:

$$\textbf{Denoising:}\quad E(x; x_0) = \|x - x_0\|^2$$
$$\textbf{Super-resolution:}\quad E(x; x_0) = \|d(x) - x_0\|^2$$
$$\textbf{Inpainting:}\quad E(x; x_0) = \|(x - x_0) \odot m\|^2$$

$d(x)$ is a downsampling operator (convolution with fixed weights), m is a binary mask denoting missing pixels,

$$\textbf{Objective:}\quad \arg\min_{\theta} E(f_\theta(z); x_0)$$

$$\textbf{Metric:}\quad PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Denoising

$$E(x; x_0) = \|x - x_0\|^2$$

Model	PSNR
Single DIP	29.22
+ Avg over iters	30.43
+ Avg over runs	31.00
CBM3D	**31.42**
Non-local means	30.26

Original

Input

Deep Img. Prior

CMB3D

Super-resolution

$$E(x; x_0) = \|d(x) - x_0\|^2$$

Model	Set5	Set14
DIP	29.9	27
Bicubic	28.43	26.05
SRResNet	**32.10**	**28.53**

Original

Bicubic

Deep Img. Prior

SRResNet

Inpainting

$$E(x; x_0) = \|(x - x_0) \odot m\|^2$$

Image	DIP	Papyan et al.
Barbara	**32.22**	28.14
Boat	**33.06**	31.44
House	**39.16**	34.58
Lena	**36.16**	35.04
Peppers	**33.05**	31.11

Original

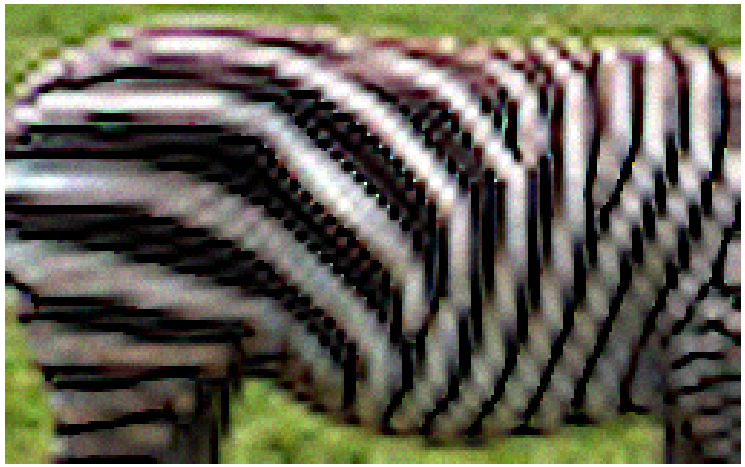
Input

Deep Img. Prior

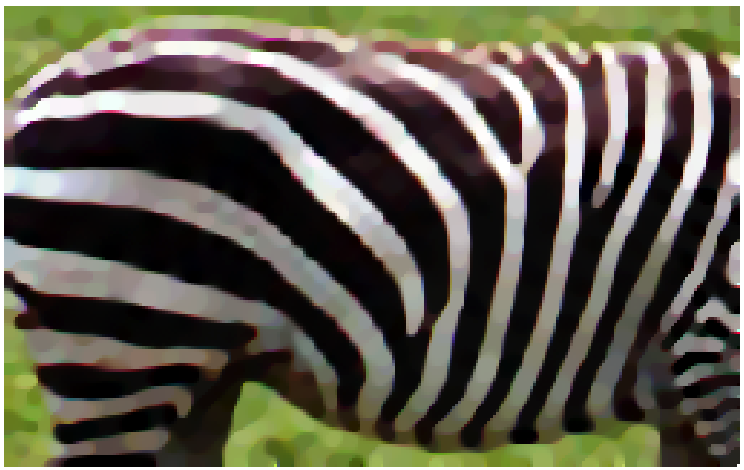
Papyan et al.

Effect of prior

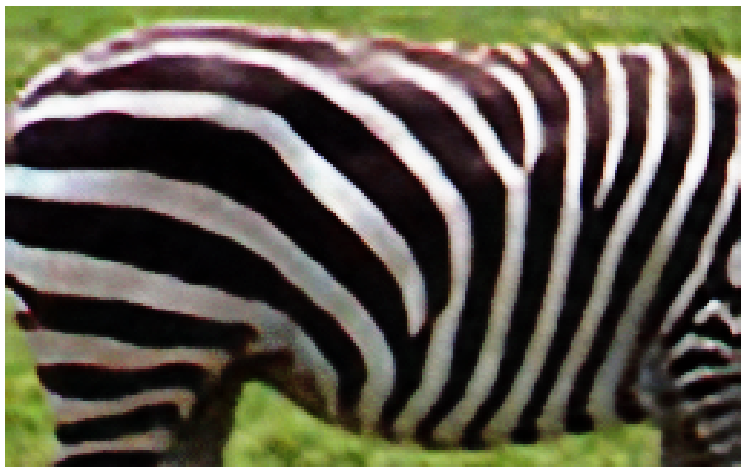
HR image



No prior



TV prior



Deep image prior

Architecture choice

Input

Encoder-decoder

ResNet

U-net

Conclusion

- Deep learning approach is successful at image generation not only because of the ability to learn
- The networks are **better hand-crafted priors**
- A great deal of image statistics are captured by the structure of a generator **independent of learning**
- Randomly initialized ConvNet works as a "Swiss knife" for image restoration problems
- While somewhat slow (several minutes on GPU), it does not require either training set or data degradation model

References I

Deep Image Prior
Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky
arXiv:1711.10925 [cs.CV]

Image denoising by sparse 3-D transform-domain collaborative filtering
Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, Karen Egiazarian
IEEE Transactions on Image Processing 16, no. 8 (2007): 2080-2095

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi
arXiv:1609.04802 [cs.CV]

References II

Convolutional Dictionary Learning via Local Processing
Vardan Papyan, Yaniv Romano, Jeremias Sulam, Michael Elad
arXiv:1705.03239 [cs.CV]

U-Net: Convolutional Networks for Biomedical Image Segmentation
Olaf Ronneberger, Philipp Fischer, Thomas Brox
arXiv:1505.04597 [cs.CV]