Demystifying — Deep Image Prior

Introduction to image restoration using deep image prior.

Pratik Katte · Follow
Published in Towards Data Science
6 min read · Feb 15, 2019

In this post, I will mainly be focusing on task of image restoration and how deep image prior can be used to solve this task.

Introduction to Image Restoration

Image restoration refers to the task of recovery of an unknown true image from its degraded image. The degradation of image may occur during image formation, transmission, and storage. This task has a wide scope of usage for satellite imaging , low-light photography and due to advancement in digital technology, computational and communication technology restoration of clean image from the degraded image is very important and hence, has evolved into a field of research which intersects with image processing, computer vision, and computational imaging.

There are mainly three tasks in image restoration:

Image Denoising:

Image denoising refers to the restoration of an image contaminated by additive noises. This is the simplest task in image restoration and therefore has been extensively studied by several technical communities.

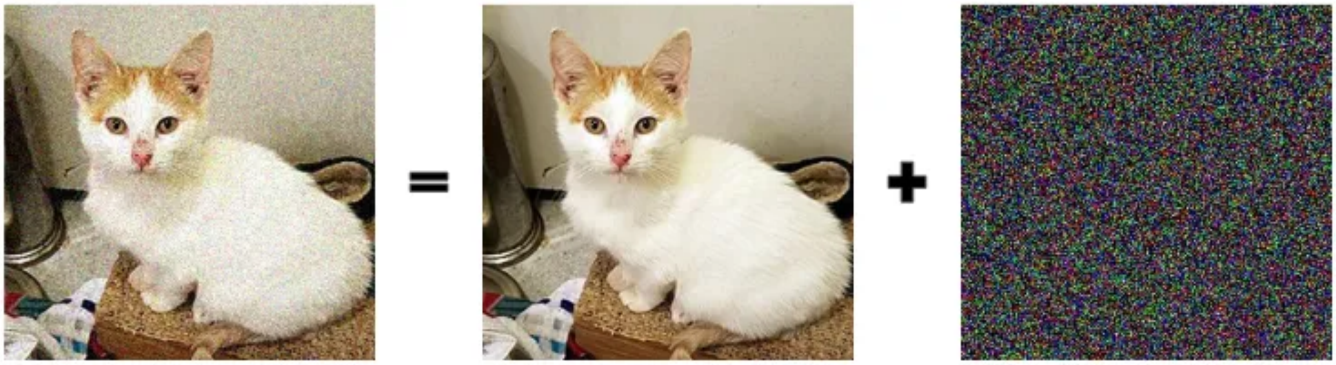

Fig.1 (left)noise add image, (center)true image, (right)Gaussian noise

2. super resolution:

super resolution refers to the process of producing a high-resolution image(or a sequence of high-resolution images) from a set of low-resolution images.

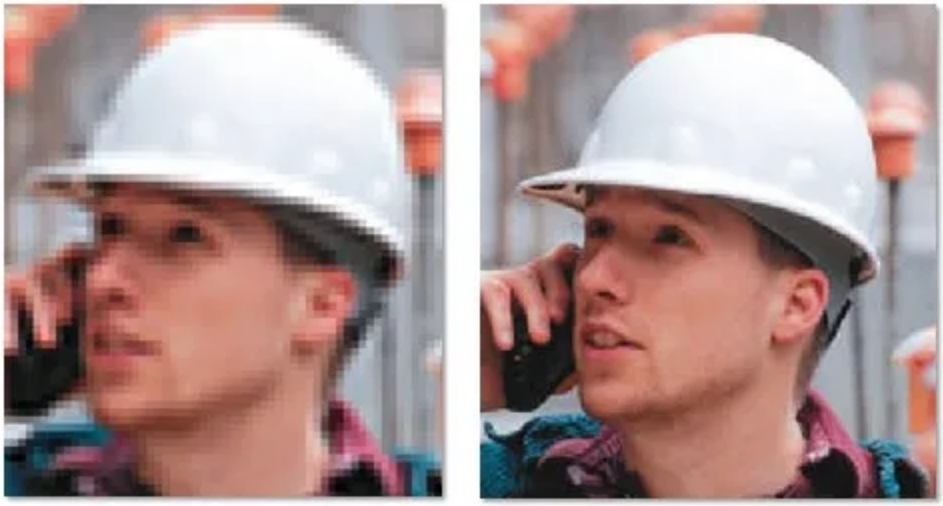

Fig2. (left)low resolution image, (right)high resolution image

3. image in-painting:

Image in-painting is the process of reconstructing the lost of deteriorated parts of images. In-painting is actually an ancient art that required humans to paint the deteriorated and lost portion of the painting. But in today's world research have come up with numerous ways to automate this task using deep convolution networks.

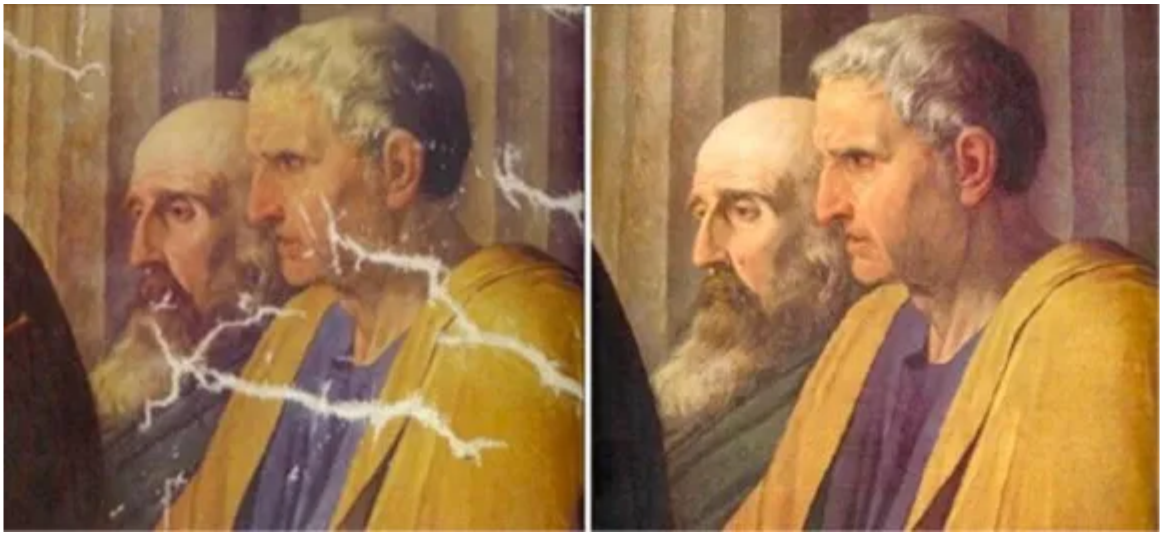

Fig. 3 (left) input, (right) output

What is Deep Image Prior?

Following the success of alexnet in image-net competition in 2012, convolution neural network has become very popular and have been used in every computer vision and image processing tasks and has been extensively used to perform inverse image reconstruction task and has achieved state-of-the-art performances.
The deep convolution networks have been successful because of their ability to learn from large amounts of image datasets. The startling paper “Deep Image Prior” by Dmitry Ulyanov showed that to solve inverse problems like image restoration, the structure of the network is sufficient and imposes a strong prior to restore the original image from the degraded image. The paper emphasizes that to perform these tasks, pretrained network or large image datasets are not required and can be performed taking degraded image into consideration only.

To perform the task of image restoration, learned-prior and explicit-prior are the two popular and majorly used by researchers.

In learned-prior is a straight forward approach to train the deep convolution network to learn about the world through the dataset, which takes in noisy image as input and clean image as desired output.
On the other hand, explicit-prior or hand-crafted prior method, is in which we embed hard constrained and teach what types of images are natural, face, etc. from

the synthesized data. It is very difficult to express constraints like natural mathematically.

In deep image prior,author tries to bridge the gap between two popular methods by constructing a new explicit prior using convolution neural network.

Let’s Go Technical …

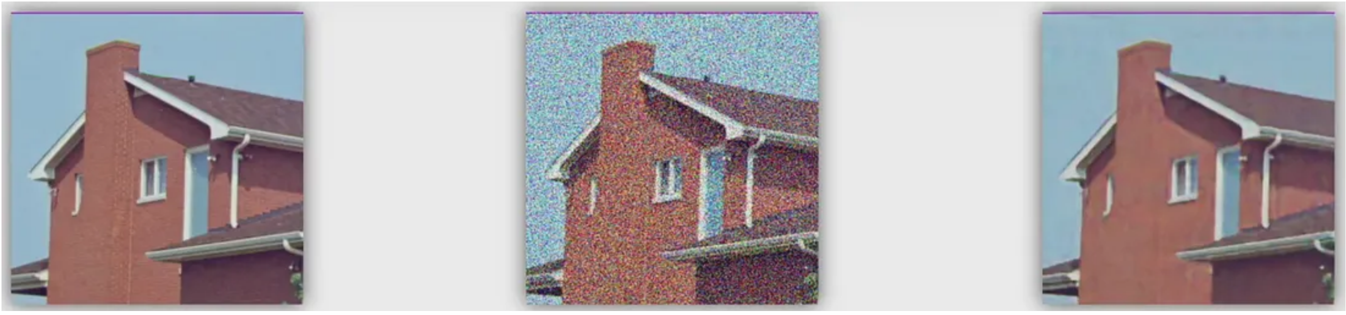

Fig.4. (left)clean image, (center)corrupted image, (left) restored image

x → *clean image*
$\dot{x}$ → *degraded image*
x^* → *restored image*

We can use maximum posteriori distribution to estimate the unobserved value from the empirical data

$$x^* = arg\ maxp(\dot{x}|x)$$

using the Bayesian rule we can express it as *likelihood * prior.*

$$p(x|\dot{x}) = \frac{p(\dot{x}|x)p(x)}{p(\dot{x})}\ \alpha\ \underbrace{p(\dot{x}|x)}_{\text{likelihood}}\underbrace{p(x)}_{\text{Prior}}$$

Instead of working with distributions separately we can can formulate the equation as optimization problem:

By applying negative algorithm to Eq. (1)

$$
\begin{aligned}
x^* &= arg\ max p(\dot{x}|x) \qquad (1)\\
&= arg\ min - log p(\dot{x}|x) - log p(x)\\
&= arg\ min\ E(x;\dot{x})\ + R(x) \quad (2)
\end{aligned}
$$

E(x;ẋ) is the data term which is negative log of the likelihood and R(x) is the image prior term which is negative log of the prior.

Now the task is to minimize Eq (2) over the image *x*. The conventional approach is to initialize *x* with random noise and then compute the gradient of the function with respect to x and traverse the image space until we converge to some point.

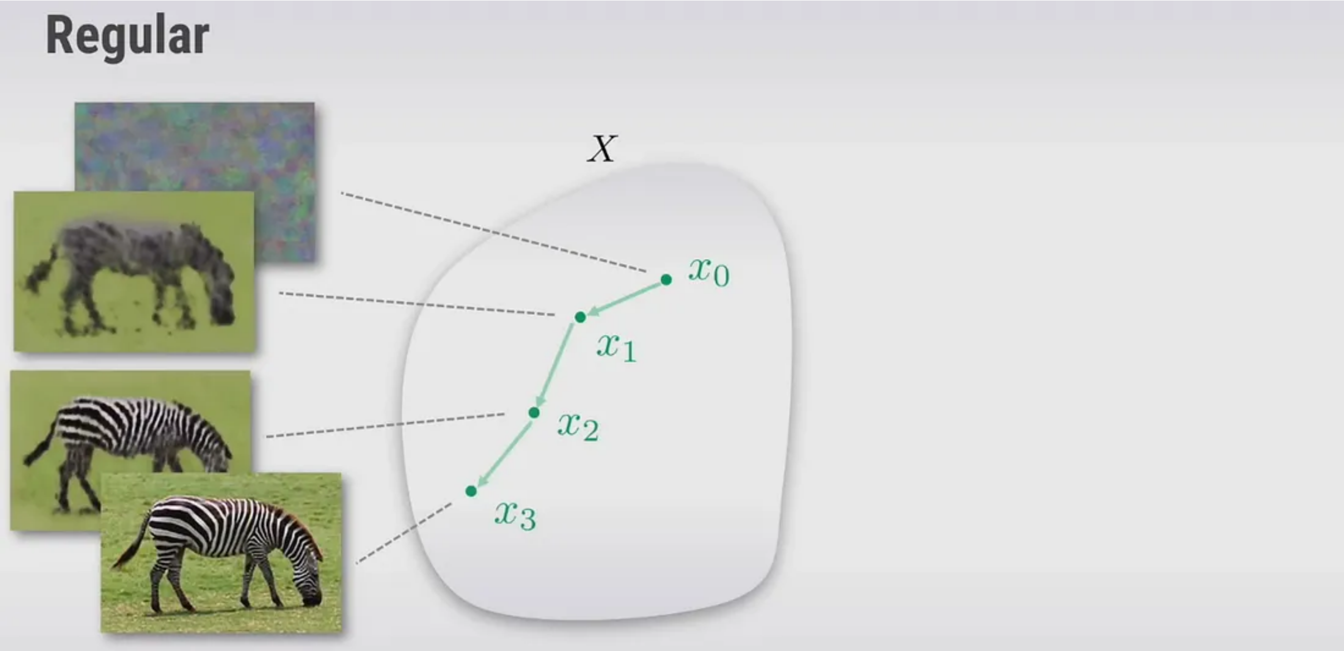

Fig. 5 visualization of regular approach

Another approach is to construct a function *g* which initializes with random θ who's output from a different space can be mapped to image x and update the θ's using gradient descent until until it converges at some point. so, instead of optimizing over the image space, we can optimize over the θ.

$$arg\ min\ E(g(\theta);\dot{x})\ + R(g(\theta))$$

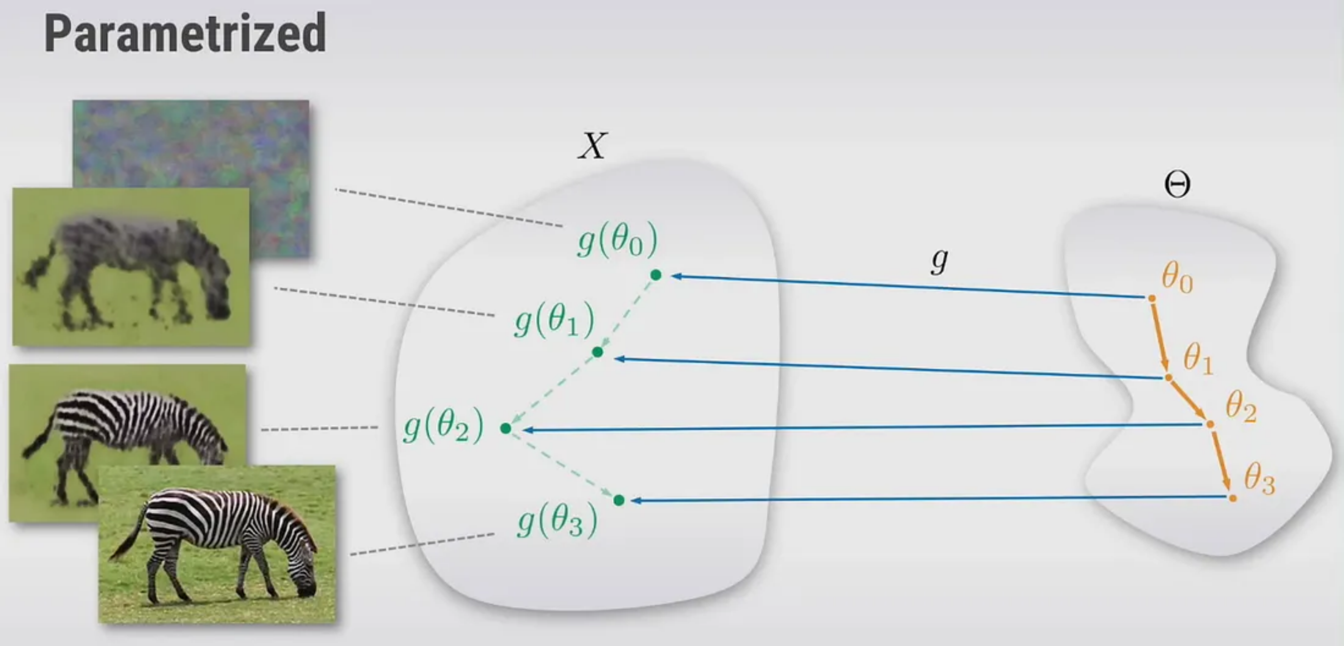

FIg. 6 Visualization of parameterized approach

But, why is this approach possible and why should we use it?. This is possible because theoretically, if g is surjective $\boldsymbol{g:\theta \mapsto x}$ (if atleast one θ maps to image x) then this optimization problem are equivalent, that is they have same solutions. But in practice, the *g* dramatically changes over how the optimization method searches the image space. We can actually treat *g* as hyper parameter and tune it. And if we observe, g(θ) acts as a prior which helps in selecting a good mapping which gives a desired output image and prevents use from getting the wrong images.
So, instead of optimizing the sum of two components. We will now optimize only the first term.

Now, the Eq 2. can be expressed as,

$$g(\theta) \equiv f_{\theta}(z)$$

$$arg \ \min_{\theta} \ E(f_{\theta}(z); \dot{x})$$

where, z is the random fixed input image and θ is randomly initialized weights which will be updated using gradient descent to get the desired output image.

But still it is not obvious why should we consider this parameterization method. Theoretically at first glance, it would seem like it would generate original noisy image. In paper, the authors conducted an experiment that showed that when gradient descent is used to optimize the network, the convolution neural network are reluctant to noisy images and descends much more quickly and easily towards naturally-looking images.

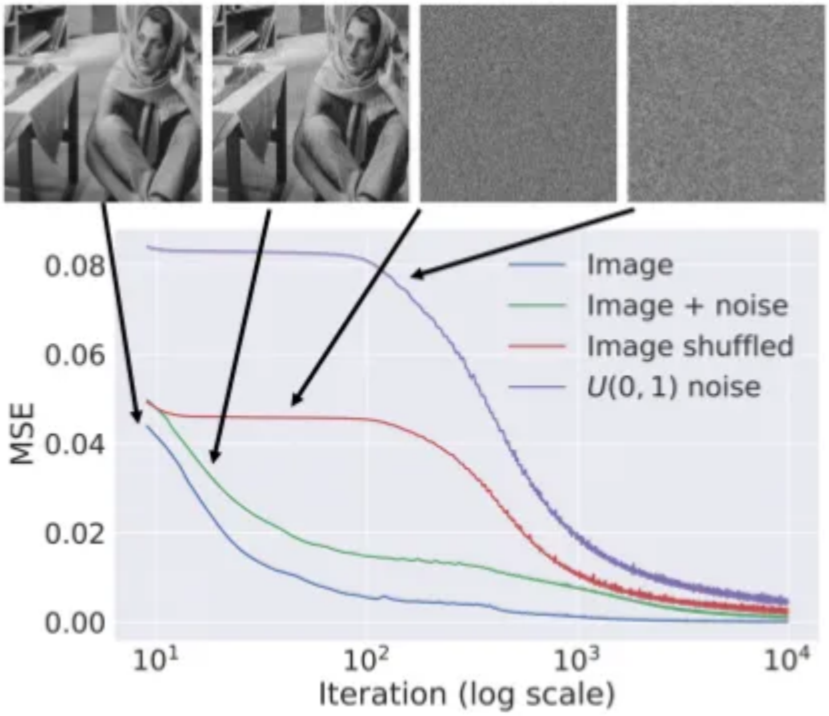

Fig. 7 Learning curves for the reconstruction task using: a natural image, the same plus i.i.d. noise, the same randomly scrambled, and white noise. Naturally-looking images result in much faster convergence, whereas noise is rejected.

Deep Image Prior Step By Step

$\dot{x}$ = *corrupted image* (observed)

1. Initialize ***z.*** : Fill the input *z by* uniform noise, or any other random image.

2. solve and optimize the function using gradient-based method.

$$\arg\min_{\theta} E(f_{\theta}(z); \dot{x})$$

$$\theta^{k+1} = \theta^{k} - \alpha \frac{\delta E(f_{\theta}(z);\dot{x})}{\delta\theta}$$

3. And finally when we find the optimal θ, we can **get the optimal image**, by just forward passing the fixed input *z* to the network with parameters θ.

$$x^{*} = f_{\theta^{*}}(z)$$

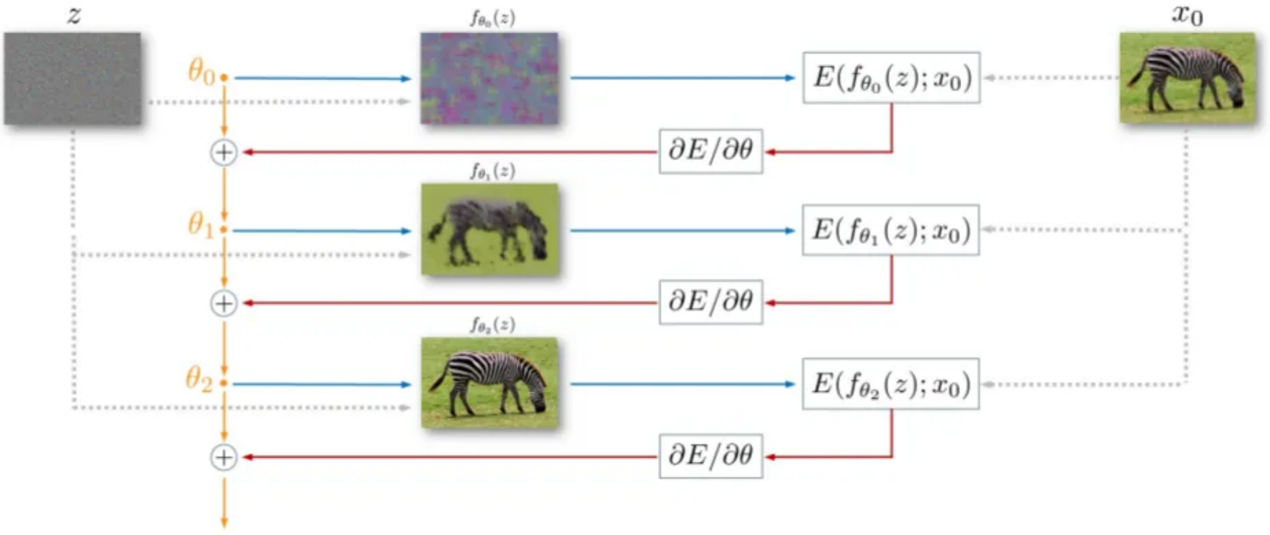
z
$f_{\theta_0}(z)$
x_0
θ_0
$E(f_{\theta_0}(z); x_0)$
$\partial E/\partial\theta$
$f_{\theta_1}(z)$
θ_1
$E(f_{\theta_1}(z); x_0)$
$\partial E/\partial\theta$
$f_{\theta_2}(z)$
θ_2
$E(f_{\theta_2}(z); x_0)$
$\partial E/\partial\theta$

Fig. 8: 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 image x = f θ (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.

Conclusion

The paper tries to shows that the approach of constructing a implicit prior inside deep convolution neural network architectures with randomized weights is well-suited for image restoration tasks. The results shown in the paper largely suggest that properly hand-crafted network architectures can be sufficient to solve image restoration task.

Machine Learning · Artificial Intelligence · Deep Image Prior · Image Restoration · Computer Vision

Follow

Written by Pratik Katte

37 Followers · Writer for Towards Data Science

Love diving deep into networks. | working towards building and deploying machine learning models for the masses.

More from Pratik Katte and Towards Data Science