Deep Image Prior

Team no: 16

Priyansh Upadhyay – 2021201090

Nilesh Jha - 2021202017

Aditya Gonnade - 2021202023

Introduction

- ConvNets trained on a vast data set of images can be called a learned prior.
- The performance of learned priors can be attributed to their capacity to learn realistic image priors from its training dataset.
- Explicit or handcrafted priors on the other hand, prior to any learning, the structure of a generator network is adequate to record a large amount of low-level image statistics.
- Aim: To create a structure of fully convolutional network that is sufficient to capture a great deal of low-level statistics prior to any learning

Problem Statement

Problem Statement

- ConvNets are typically trained on huge image datasets, one may believe that their superior performance is due to the fact that they learn realistic data priors from examples, however this explanation is insufficient.
- Not all image priors must be learned from data, instead, a great deal of image statistics are captured by the structure of generator ConvNets, independent of learning.
- This is especially true for the statistics required to solve certain image restoration problems, where the image prior must supplement the information lost in the degradation processes.

Approach

Approach

- The network weights are used to parametrize the restored images in this technique.
- The weights are randomly initialized and fitted to a specific degraded image under a task-dependent observation model.
- In this way, the single degraded input image and the handmade network structure utilised for reconstruction are the only sources of information for reconstruction.

Main idea

In image restoration problems the goal is to recover original image x having a corrupted image x_0. Such problems are often formulated as an optimization task:

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

where $E(x; x_0)$ is a *data term* and $R(x)$ is an *image prior*. The data term $E(x; x_0)$ is usually easy to design for a wide range of problems, such as super-resolution, denoising, inpainting, while image prior $R(x)$ is a challenging one. Today's trend is to capture the prior $R(x)$ with a ConvNet by training it using large number of examples.

We first notice, that for a surjective $g : \theta \mapsto x$ the following procedure in theory is equivalent to (1):

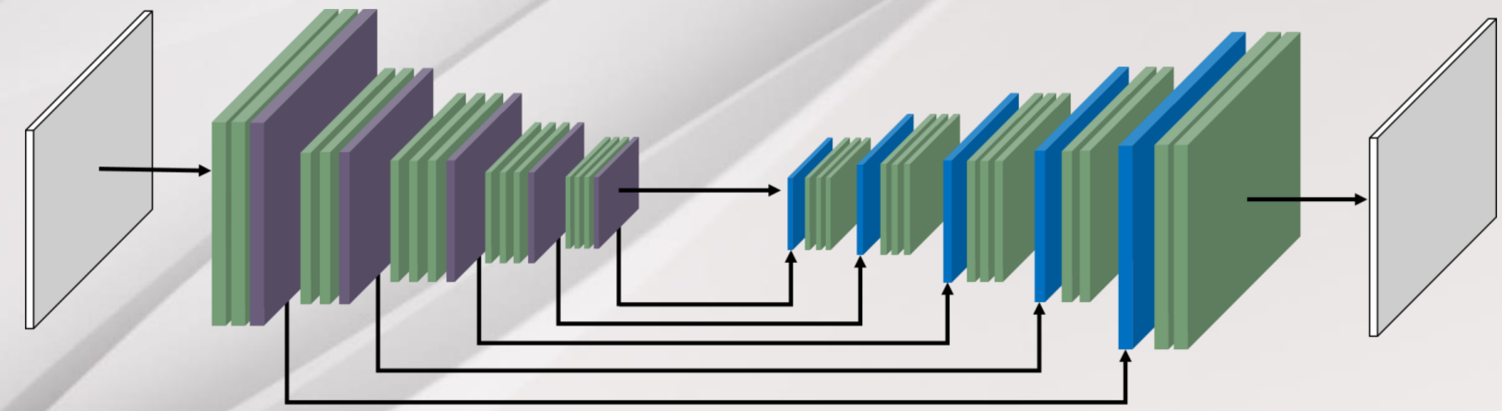
$$\min_\theta E(g(\theta); x_0) + R(g(\theta))\,.$$

In practice g dramatically changes how the image space is searched by an optimization method. Furthermore, by selecting a "good" (possibly injective) mapping g, we could get rid of the prior term. We define $g(\theta)$ as $f_\theta(z)$, where f is a deep ConvNet with parameters θ and z is a fixed input, leading to the formulation

$$\min_\theta E(f_\theta(z); x_0)\,.$$

Here, the network f_θ is initialized randomly and input z is filled with noise and fixed.

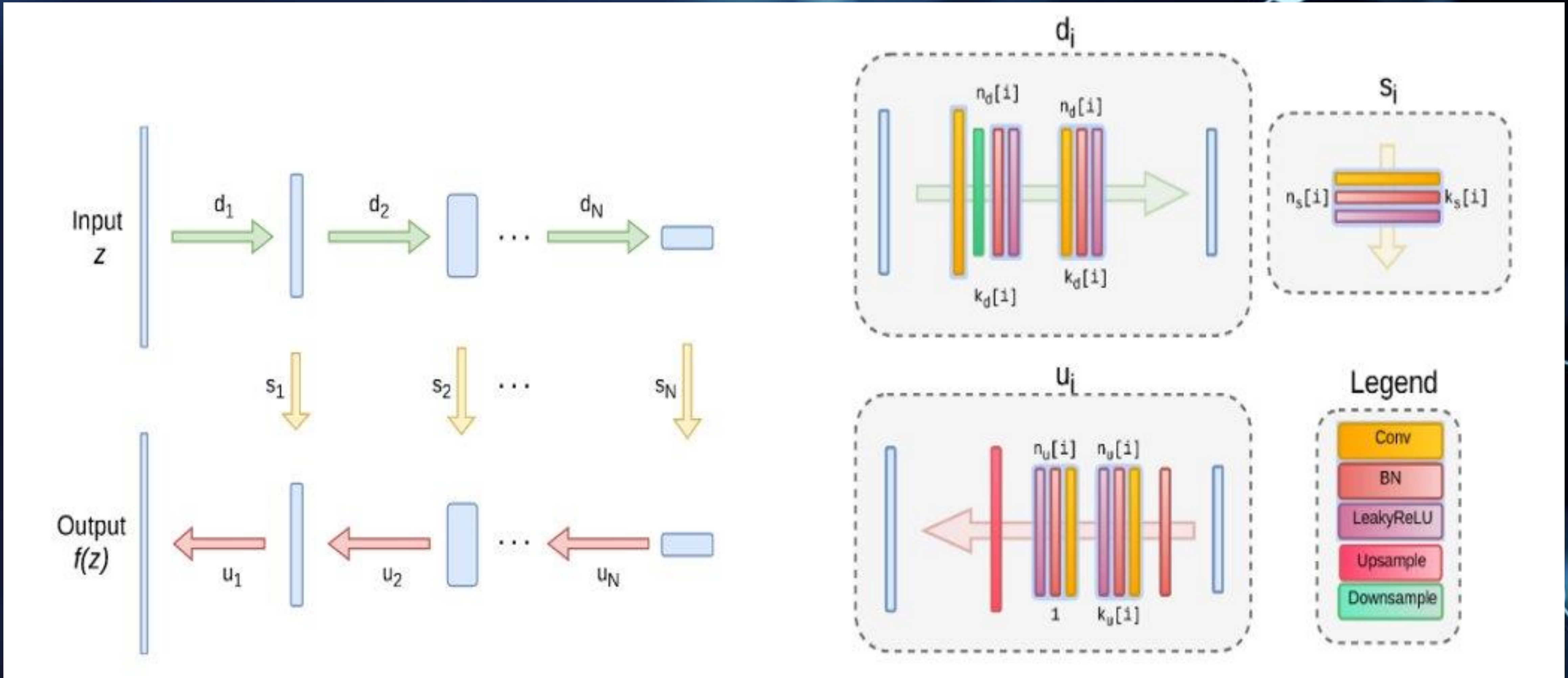
In other words, **instead of searching for the answer in the image space we now search for it in the space of neural network's parameters**. We emphasize that we never use a pretrained network or an image database. Only corrupted image x_0 is used in the restoration process.



Hourglass Model

- The network consists of a contracting path and an expansive path, which gives it the u-shaped architecture.
- The contracting path is a conventional convolutional network, consisting of convolutions applied repeatedly, each followed by a rectified linear unit (ReLU) .
- The expansive pathway combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path.

Architecture



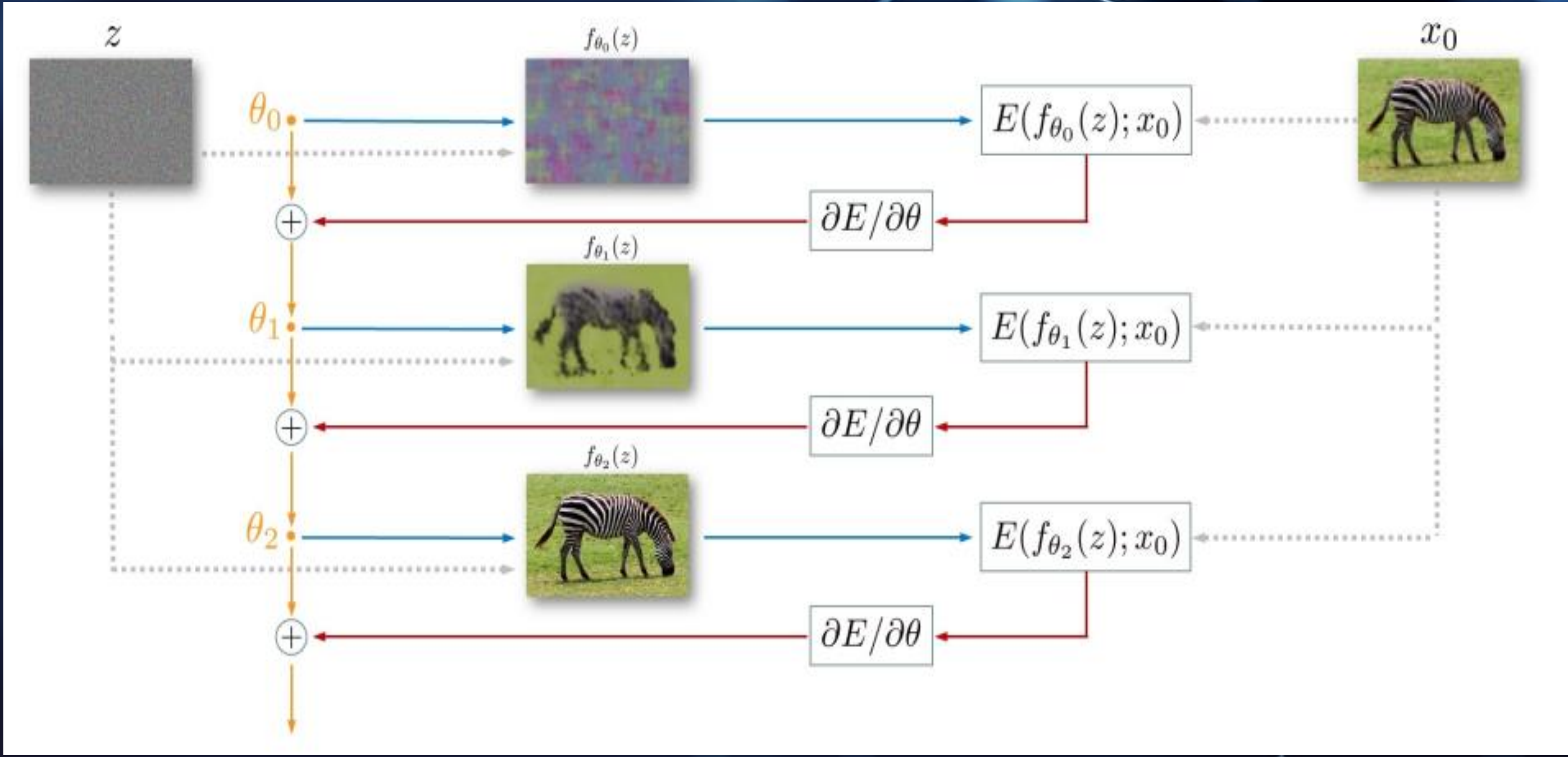
Learning Parameters

Deep Image Prior step by step

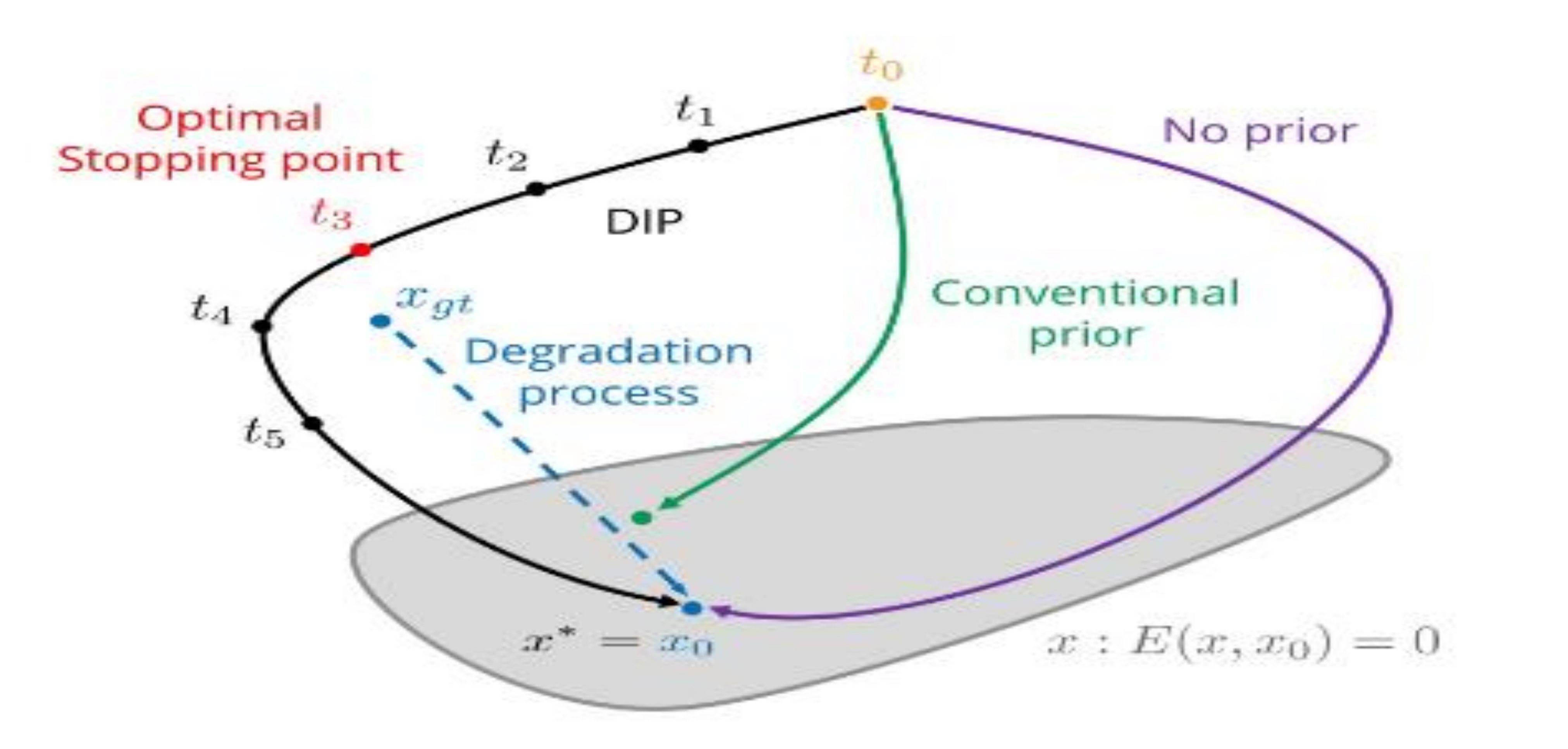
- $\hat{x}$ – Corrupted image (observed)

1. **Initialize** z
   - For example fill it with uniform noise U(-1, 1)
2. **Solve**
   $$\arg\min_{\theta} E(f_{\theta}(z); \hat{x}))$$
   - With your favorite gradient-based method
   $$\theta^{k+1} = \theta^k - \alpha \frac{\partial E(f_{\theta}(z); \hat{x})}{\partial \theta}$$
3. **Get the solution**
   $$x^* = f_{\theta^*}(z)$$

Learning Parameters



Learning Parameters



Applications

Denoising

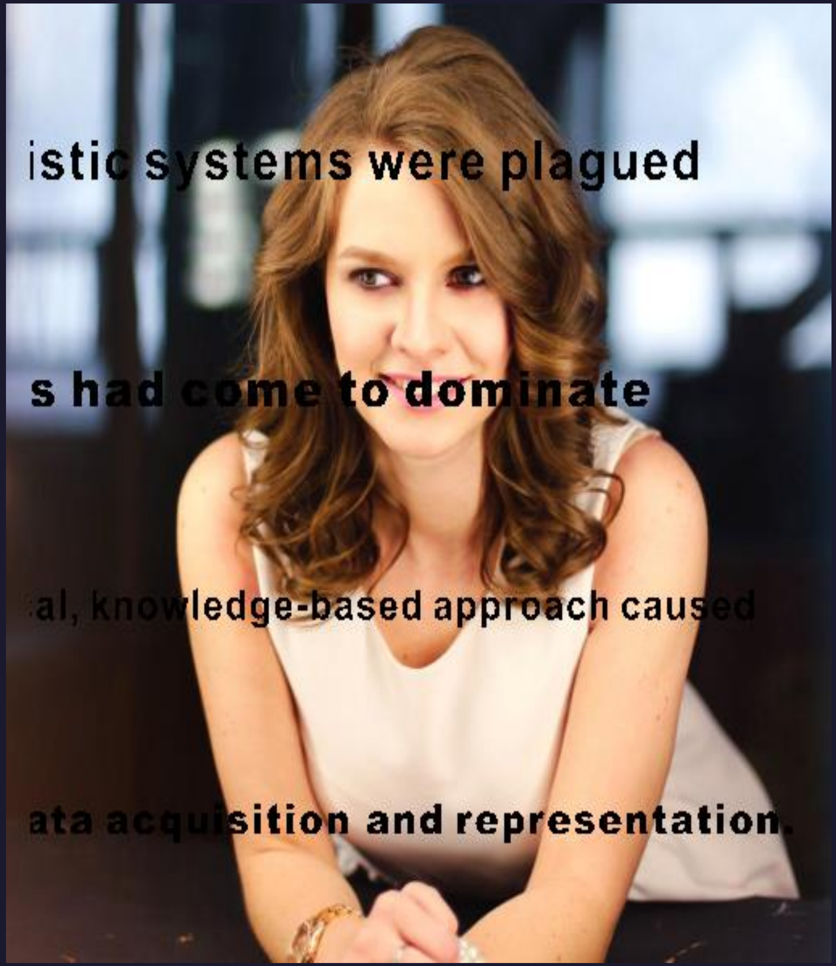
$$E(x; x_0) = \|x - x_0\|^2$$

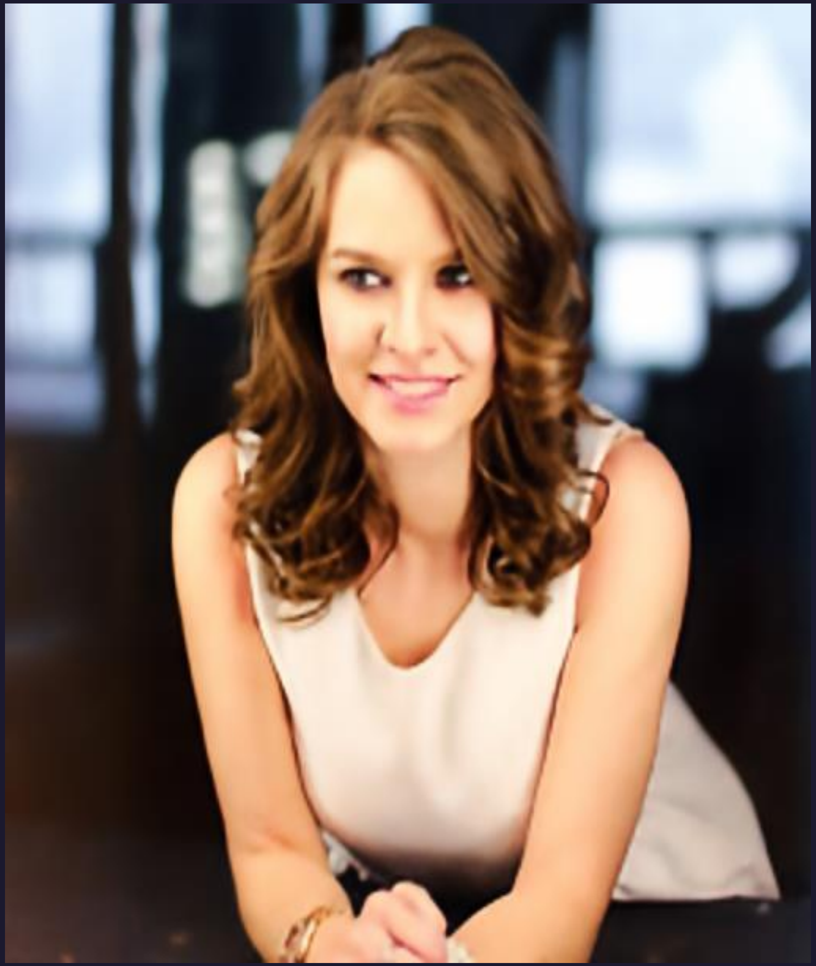
Parameters

- nu = nd = [128, 128, 128, 128, 128]
- ku = kd = [3, 3, 3, 3, 3]
- ns = [4, 4, 4, 4, 4]
- ks = [1, 1, 1, 1, 1]
- σp = 30
- num iter = 5000
- LR = 0.01

Inpainting

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





Parameters

- nu = nd = [128, 128, 128, 128, 128]
- ku = kd = [3, 3, 3, 3, 3]
- ns = [4, 4, 4, 4, 128]
- ks = [1, 1, 1, 1, 1]
- σp = 30
- num iter = 4000
- LR = 0.01

Super-resolution

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

Parameters

- nu = nd = [128, 128, 128, 128, 128]
- ku = kd = [3, 3, 3, 3, 3]
- ns = [4, 4, 4, 4, 4]
- ks = [1, 1, 1, 1, 1]
- σp = 30
- num iter = 6000
- LR = 0.01

Thank You