

# Deep Image Prior

Павел Юрлов

Высшая школа экономики  
Факультет компьютерных наук

8 февраля 2019

# План

- 1 Введение
- 2 Описание модели
- 3 Применение
  - Denoising
  - Super-resolution
  - Inpainting
  - Natural pre-image

# Введение

# Свёрточные сети

- Глубокие свёрточные сети показывают отличные результаты в задачах обратной реконструкции изображений и вообще в задачах генерации изображений
- Они почти всегда обучаются на больших датасетах
- Объясним ли их успех выучиванием image priors из данных?

# Альтернативное объяснение

Большое число сведений об изображении собирается самой структурой сети независимо от обучения, особенно сведений, необходимых при решении задач восстановления изображений, где image priors необходимы для интегрирования потерянной информации

# Идея

- Вместо обучения на большом датасете, необученная сеть получает единственное деградированное изображение
- Веса сети, инициализируемые случайно, служат параметризацией восстановленного изображения

# Идея

Восстановление изображения — задача условной генерации, требующая лишь одно входное деградированное изображение и «ручную» структуру сети

# Описание модели

# Задача генерации изображения

- Глубокая генеративная сеть является параметризованной функцией:

$$x = f_{\theta}(z), x \sim p(x)$$

- В нашем случае:

$$x \sim p(x|x_0),$$

где  $x_0$  — деградированное изображение

# Prior

- Задача: исследовать prior, имплицитно найденный выбором определённой структуры сети до того, как её параметры будут обучаться
- Сеть — параметризация  $x = f_\theta(z)$  изображения  $x \in \mathbb{R}^{3 \times H \times W}$ , где  $z \in \mathbb{R}^{C' \times H' \times W'}$  — шум, а  $\theta$  — параметры сети

# Обратные задачи: denoising, super-resolution, inpainting, etc.

$$\min_x E(x, x_0) + R(x) \quad (1)$$

- $x_0$  — деградированное изображение
- $x$  — искомое изображение
- $E(x, x_0)$  — близость/ошибка, зависит от конкретной задачи
- $R(x)$  — регуляризатор (например, полная вариация), содержащий сведения об image prior (т.е. принимающий меньшие значения при определённых  $x$ )

# Замена регуляризатора

- Вместо  $R(x)$ , имплицитный prior:

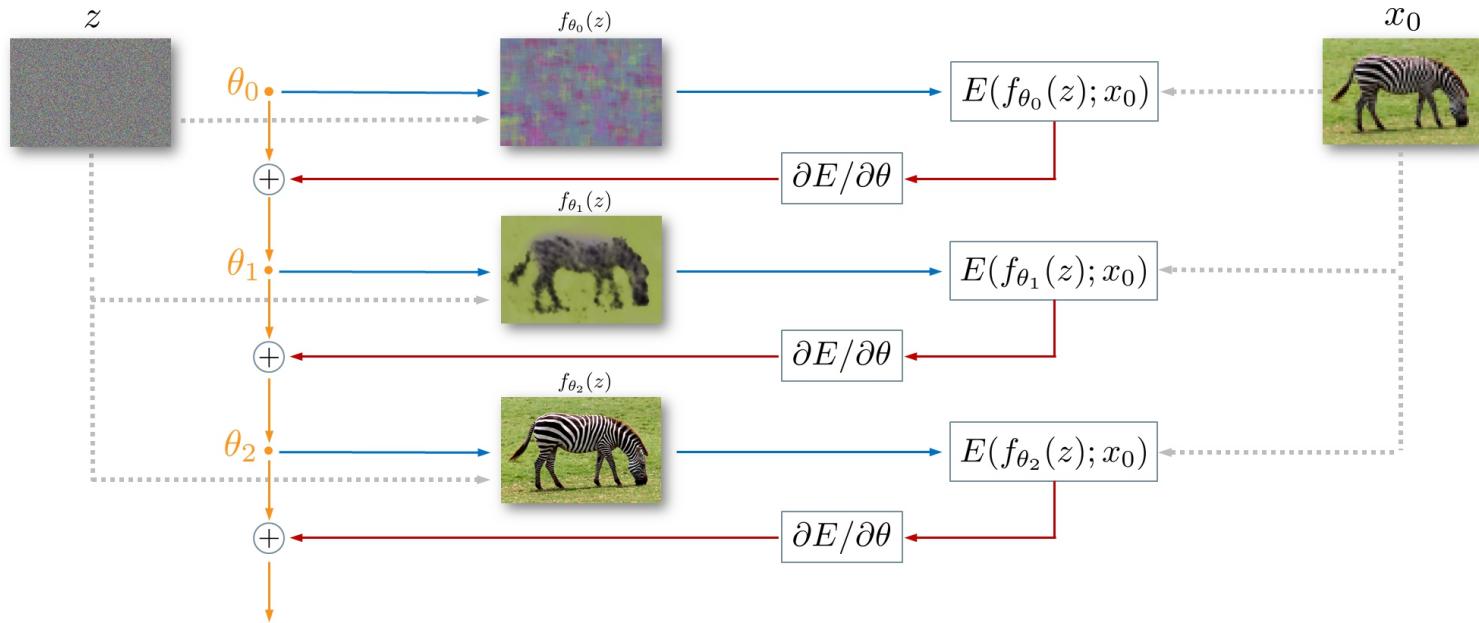
$$\theta^* = \arg \min_{\theta} E(f_{\theta}(z), x_0), \quad x^* = f_{\theta^*}(z) \quad (2)$$

- (2) в терминах (1):

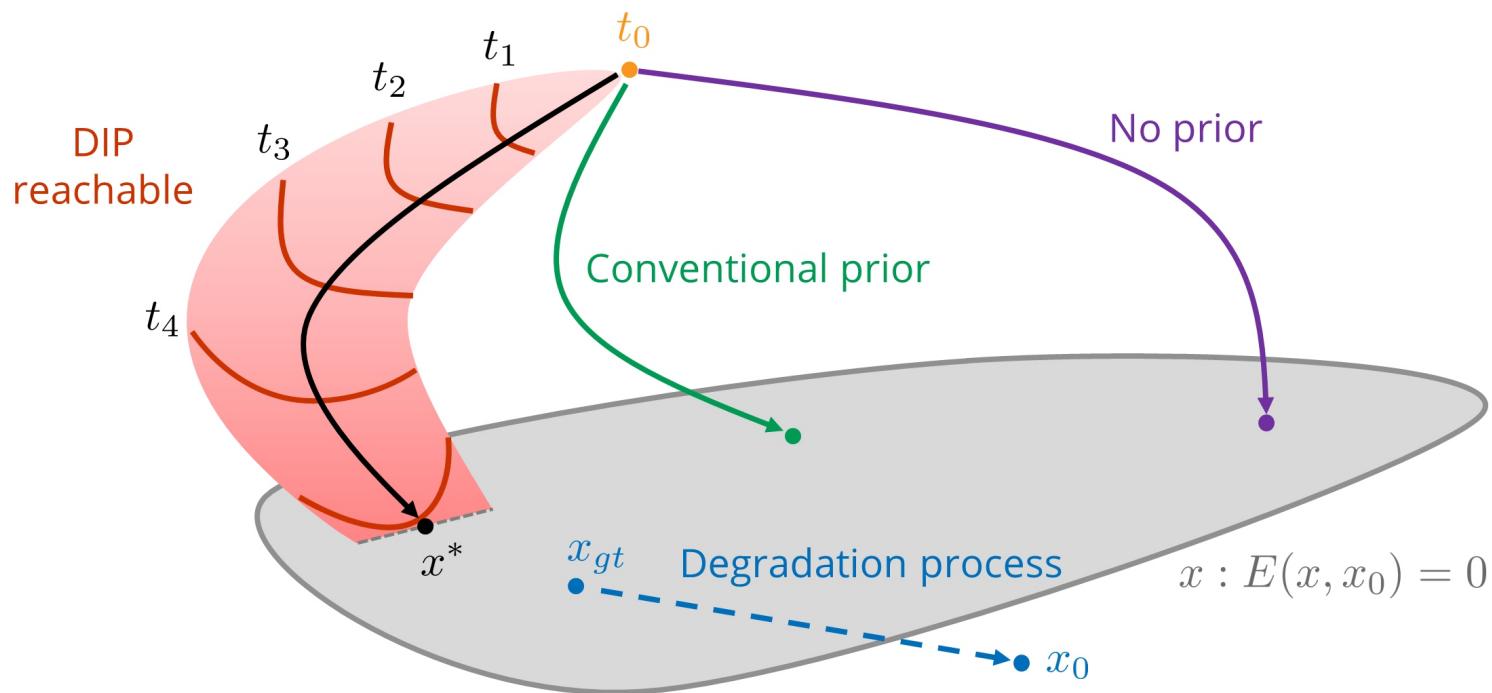
$$R(x) = \begin{cases} 0 & x \text{ может быть сгенерирован} \\ & \text{сетью определённой архитектуры} \\ +\inf & \text{иначе} \end{cases}$$

- $\theta^*$  находится градиентным спуском из случайной параметризации  $\theta_0$

# Нахождение $\theta^*$ градиентным спуском



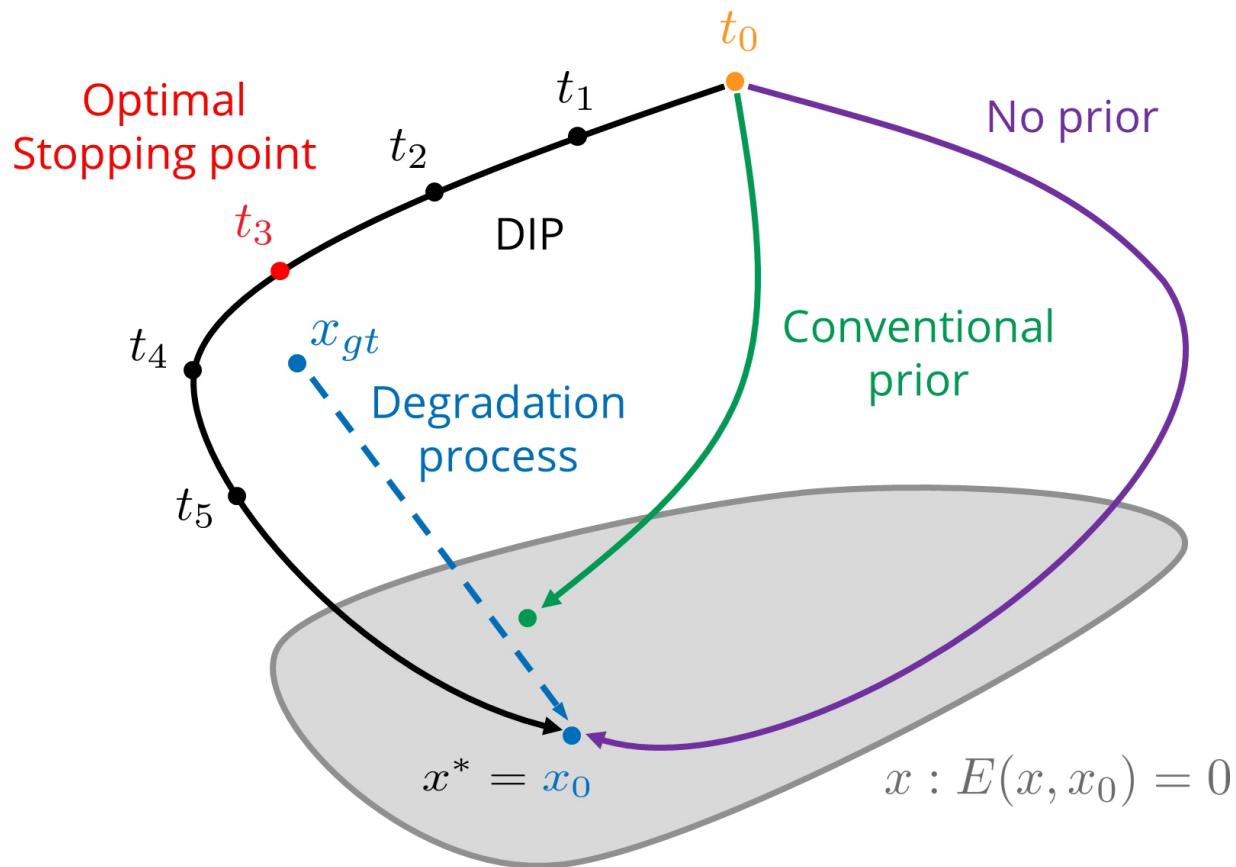
# Схема нахождения $x^*$



# Влияние параметризации $\theta$

- Разве может  $f_\theta$  вообще быть prior?
- Да, могут найтись параметры  $\theta$ , позволяющие «восстановить» почти любой  $x$
- Но архитектура влияет на процесс поиска в пространстве решений, «сопротивляясь» плохим решениям и быстрее приближаясь к естественным изображениям

# Оптимальная траектория проходит мимо $x^*$



# Объяснение

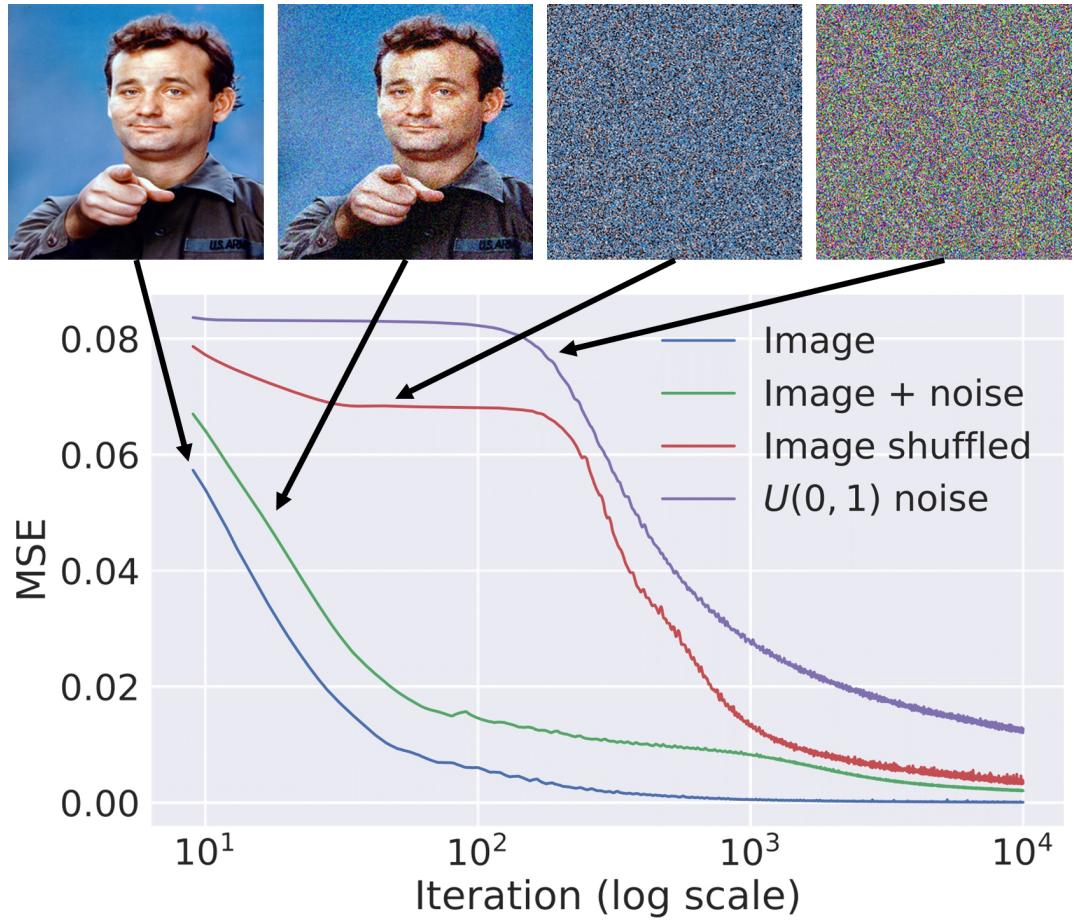
- Задача простой генерации:  
по  $x_0$  найти параметры  $\theta^*$ , воспроизводящие  $x_0$
- Подставляя

$$E(x, x_0) = \|x - x_0\|^2 \quad (3)$$

в (2), получаем

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

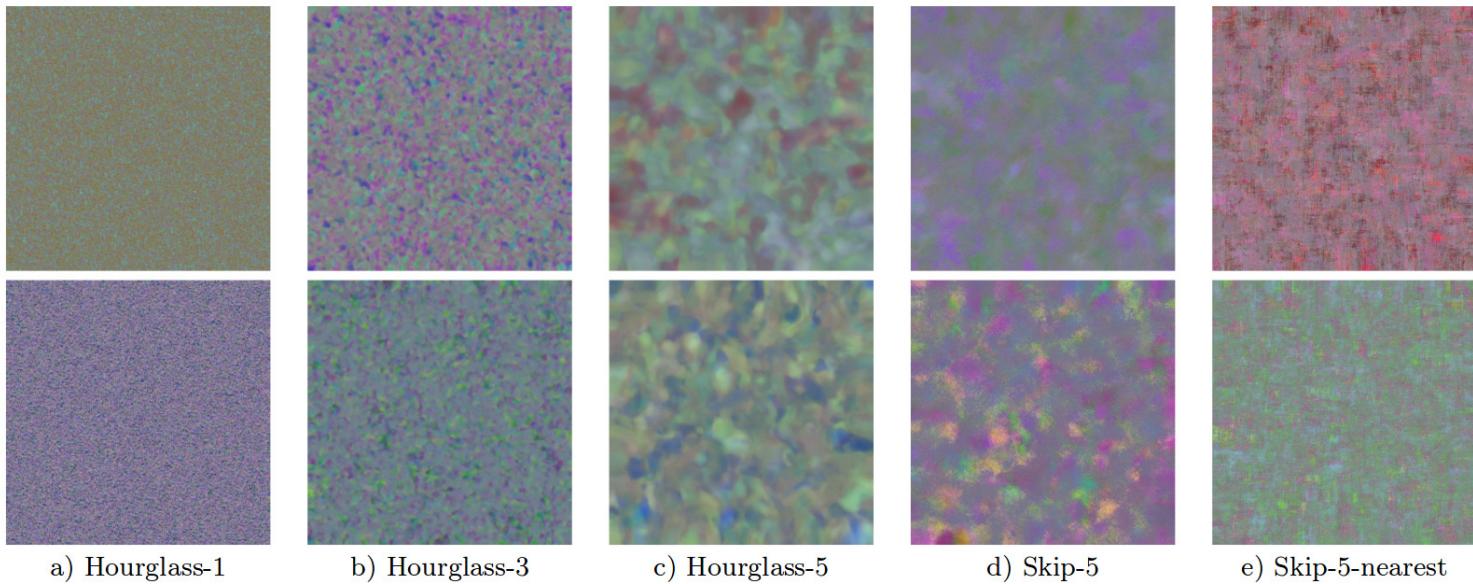
- Различные  $x_0$ : естественные изображения и шум
- В случае шума ошибка снижается намного медленнее (см. след. слайд)
- Поэтому иногда выгодно ограничить число шагов (см. пред. слайд)



# «Сэмплинг» из deep image prior

- Prior в (2) является имплицитным, поэтому «нормального» распределения на  $x$  он не даёт
- Сымитируем сэмплы —  $f_\theta(z)$  случайных параметров  $\theta$
- Получатся начальные точки оптимизации до обучения на данных
- Полученные изображения далеки от независимого случайного шума и обладают определённой внутренней схожестью

# «Сэмплы»



- В каждом столбце 2 изображения  $f_\theta(z)$  из одинаковой архитектуры, одинакового шума  $z$  и разных случайных  $\theta$
- Столбцы отличаются архитектурой сетей

# Используемая архитектура

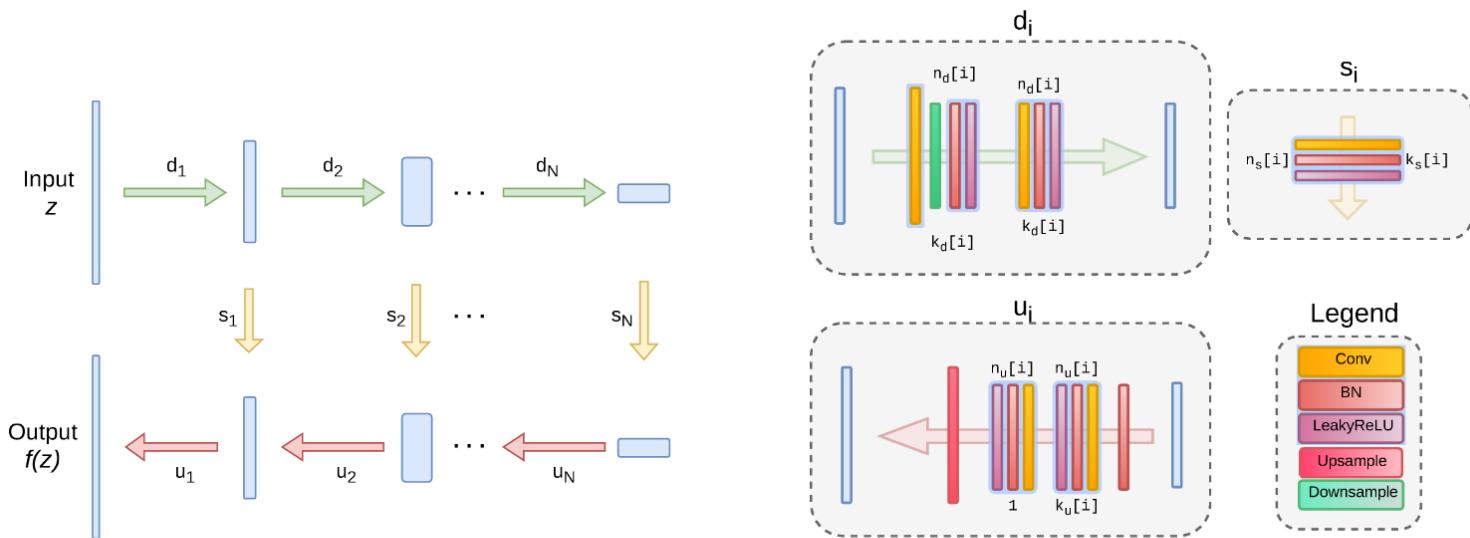


Figure 1: **The architecture used in the experiments.** We use “hourglass” (also known as “decoder-encoder”) architecture. We sometimes add skip connections (yellow arrows).  $n_u[i], n_d[i], n_s[i]$  correspond to the number of filters at depth  $i$  for the upsampling, downsampling and skip-connections respectively. The values  $k_u[i], k_d[i], k_s[i]$  correspond to the respective kernel sizes.

# Применение

# Denoising

- Восстановить чистое изображение  $x$  из зашумлённого  $x_0 = x + \epsilon$ , где по умолчанию распределение  $\epsilon$  неизвестно
- Следуем в соответствии с (3) и (4):

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

# Denoising

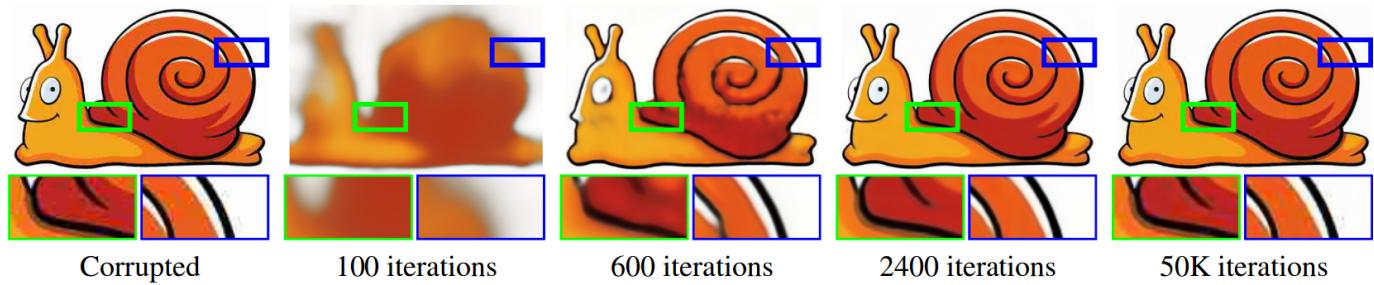


Figure 3: **Blind restoration of a JPEG-compressed image.** (*electronic zoom-in recommended*) Our approach can restore an image with a complex degradation (JPEG compression in this case). As the optimization process progresses, the deep image prior allows to recover most of the signal while getting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

# Super-resolution

- Дано изображение  $x_0 \in \mathbb{R}^{3 \times H \times W}$ , надо найти  $x \in \mathbb{R}^{3 \times tH \times tW}$
- $E(x, x_0) = \|d(x) - x_0\|^2$ ,  
где  $d(\cdot) : \mathbb{R}^{3 \times tH \times tW} \rightarrow \mathbb{R}^{3 \times H \times W}$

# Сравнение методов



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 [18], SRResNet [19]) which utilize networks trained from large datasets.

# Inpainting

- Дано изображение  $x_0$  с пропущенными пикселями в соответствии с маской  $m \in \{0, 1\}^{H \times W}$ , надо восстановить пропущенный пиксели
- $E(x, x_0) = \|(x - x_0) \circ m\|^2$

# Сравнение моделей

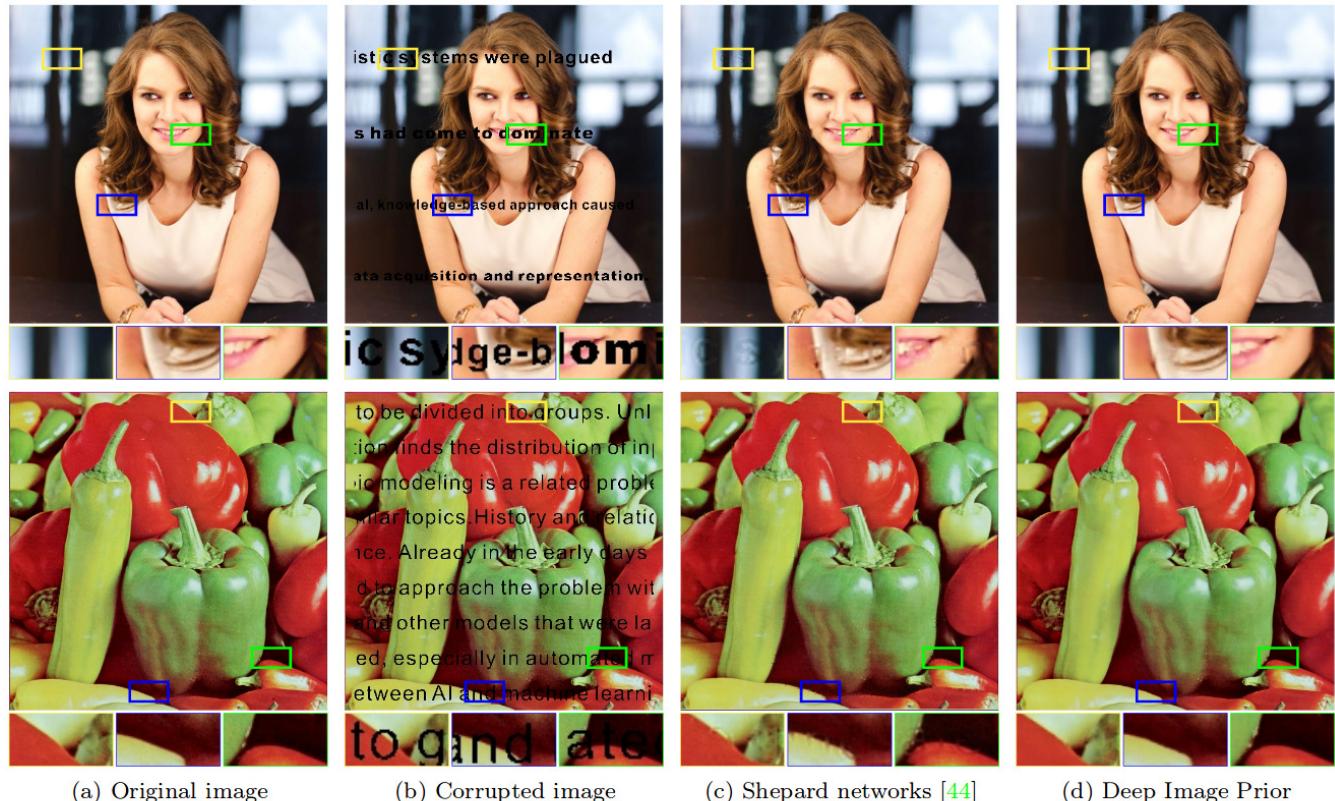


Fig. 11: Comparison with Shepard networks [44] on text the inpainting task. Even though [44] utilizes learning, the images recovered using our approach look more natural and do not have halo artifacts.

# Сравнение моделей 2

	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	<b>32.22</b>	<b>33.06</b>	<b>39.16</b>	<b>36.16</b>	<b>33.05</b>	<b>29.8</b>	<b>32.52</b>	<b>32.84</b>	<b>32.77</b>	<b>32.20</b>	<b>34.54</b>

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

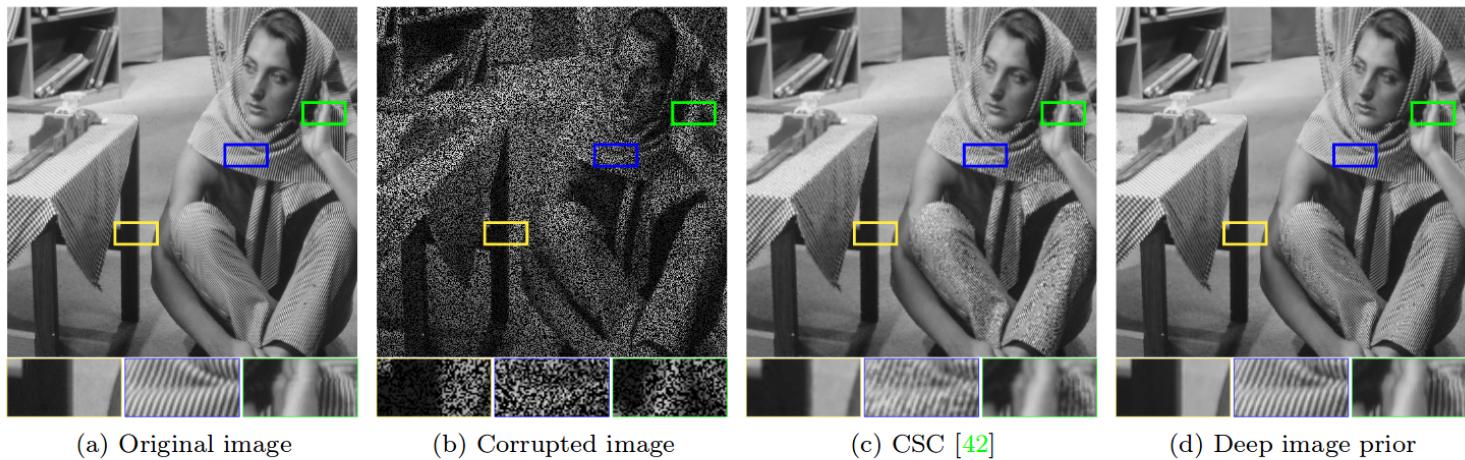


Fig. 13: Comparison with convolutional sparse coding (CSC) [42] on inpainting 50% of missing pixels. Our approach recovers a natural image with more accurate fine details than convolutional sparse coding.

# Natural pre-image

- Инструмент, позволяющий изучать функции с потерями данных (lossy functions) над естественными изображениями
- Например,  $\Phi$  — первые несколько слоёв нейронной сети, классифицирующей изображения
- Прообраз — множество  $\Phi^{-1}(\Phi(x_0)) = \{x \in \mathcal{X} : \Phi(x) = \Phi(x_0)\}$  изображений, имеющих то же представление, что и  $x_0$
- Это множество несёт сведения о том, что теряется при преобразовании, а что остаётся

# Решение задачи

- Минимизация  $E(x, x_0) = \|\Phi(x) - \Phi(x_0)\|^2$
- Но очень важен регуляризатор, так как могут получиться неестественные изображения

# Сравнение методов

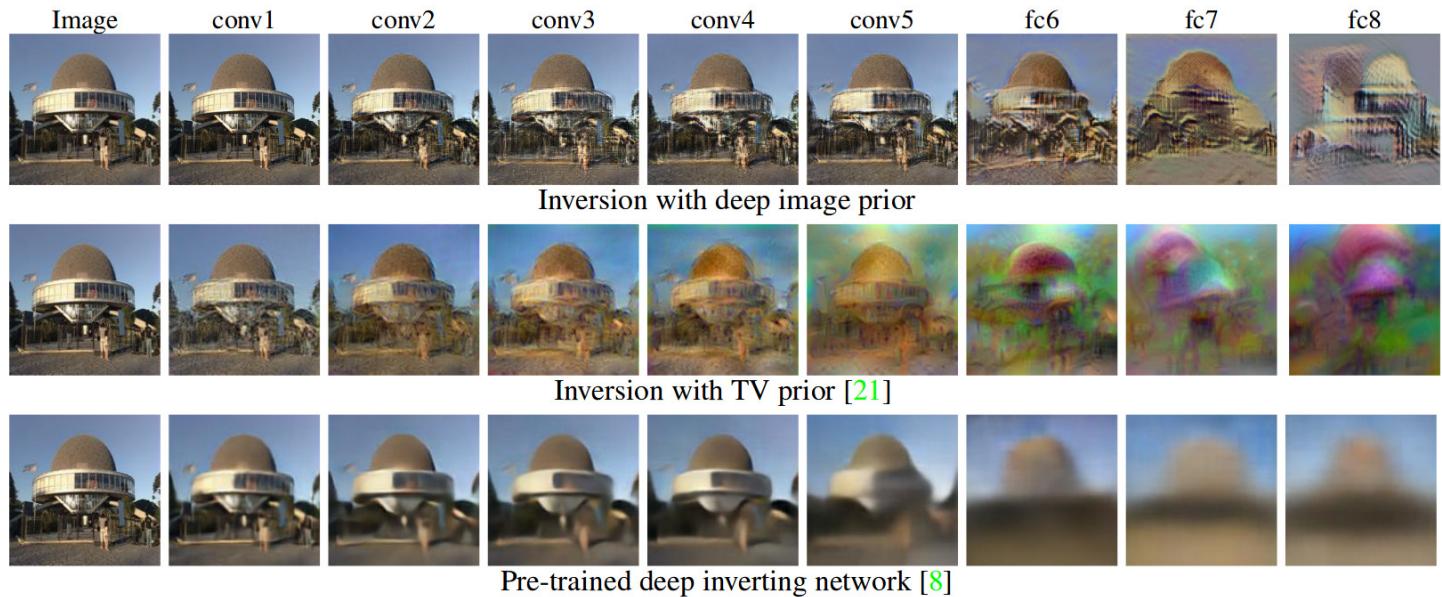


Figure 9: **AlexNet inversion.** Given the image on the left, we show the natural pre-image obtained by inverting different layers of AlexNet (trained for classification on ImageNet ISLVRC) using three different regularizers: the Deep Image prior, the TV norm prior of [21], and the network trained to invert representations on a hold-out set [8]. The reconstructions obtained with the deep image prior are in many ways at least as natural as [8], yet they are not biased by the learning process.

# Выводы

- Независимо от обучения на внешней выборке, чрезвычайно важна сама архитектура нейронной сети
- Плюсы: подход не требует моделирования процесса деградации и предварительного обучения
- Минус: низкая скорость
- Следствие: важна разработка новых архитектур

# Источники

[https://dmitryulyanov.github.io/deep\\_image\\_prior](https://dmitryulyanov.github.io/deep_image_prior)