Lab 2: Deep Image Prior

Lab Objective:

In this lab, you will be asked to reproduce the experiments of the paper "deep image prior".

Important Date:

1. Experiment Report Submission Deadline: 4/10 (Tue) 12:00
2. Demo date: 4/10 (Tue)

Requirements:

- Experiment of figure 2
- Denoising
- Super-resolution

Environment:

Download image:

Sample Code:

Official code:
https://github.com/DmitryUlyanov/deep-image-prior

Lab Description:

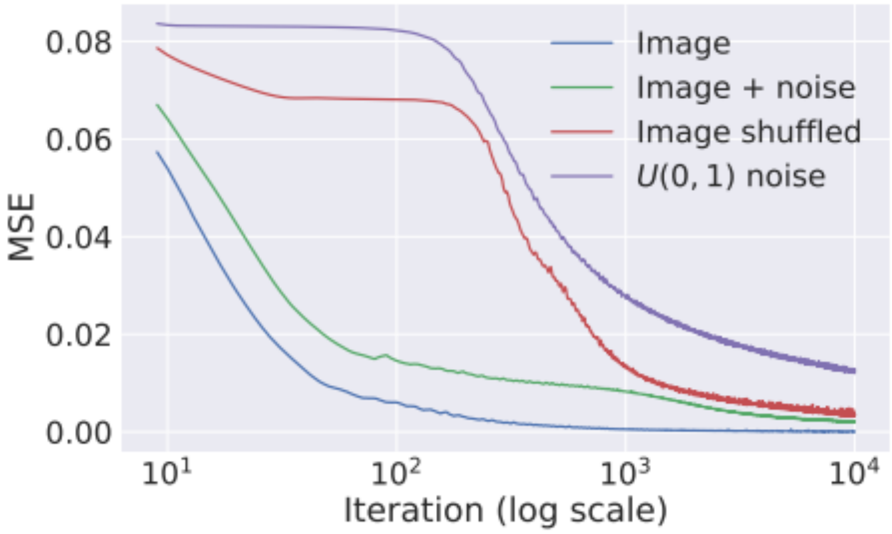
- Deep image prior
  - the structure of a generator network is sufficient to capture a great deal of low-level image statistics prior to any learning
  - a **randomly-initialized** neural network can be used as a handcrafted prior with excellent results in standard inverse problems such as denoising, superresolution, and inpainting
  - Parameterization image x
    - $x = f_{\theta}(z)$
  - Inverse problems
    - Denoising, super-resolution, inpainting

$$x^* = \min_x E(x; x_0) + R(x),$$

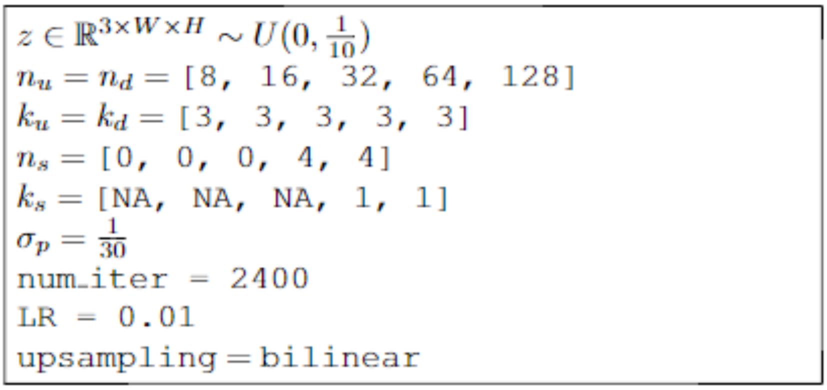
- Let R(x) is the prior captured by neural network

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

- Requirement 1
  - A parametrization with high noise impedance



  - Network architecture



$z \in \mathbb{R}^{3\times W\times H} \sim U(0, \frac{1}{10})$

$n_u = n_d =$ [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]

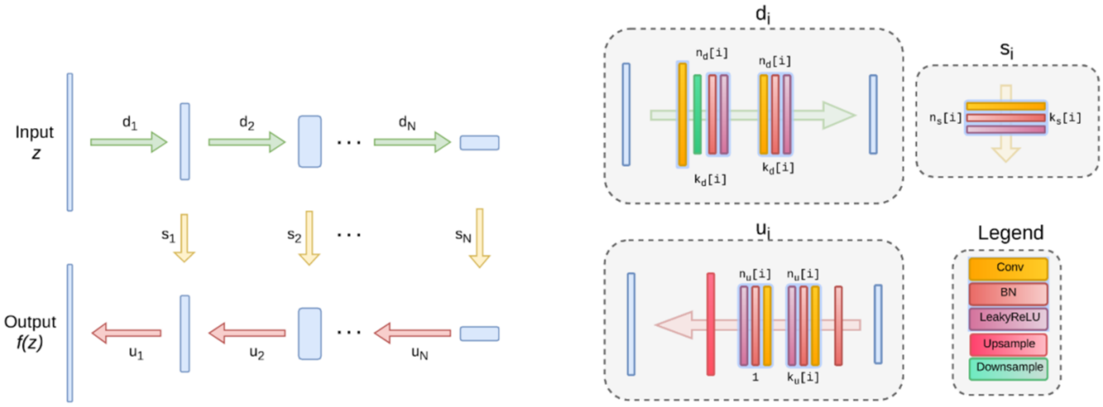
$\sigma_p = \frac{1}{30}$

num_iter = 2400

LR = 0.01

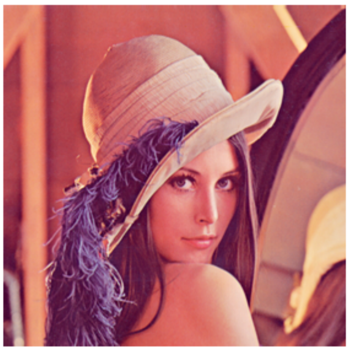
upsampling = bilinear

Optimizer: Adam

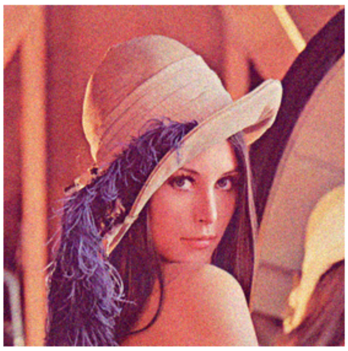


- Requirement 2
  - Blind image denoising

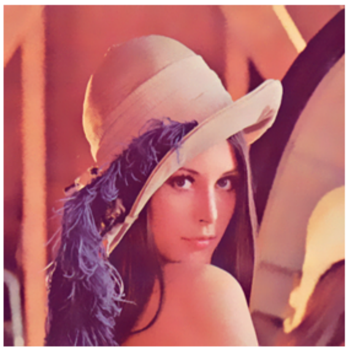
Ground Truth



Input image

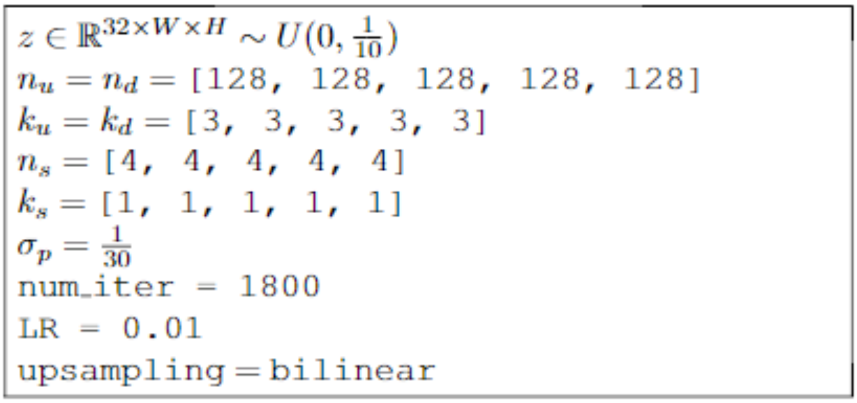


output image



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

  - Network 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 = 1800

LR = 0.01

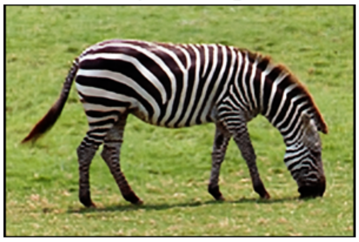
upsampling = bilinear

Optimizer: Adam

- Requirement 3
  - Bilnd super-resolution

Ground Truth

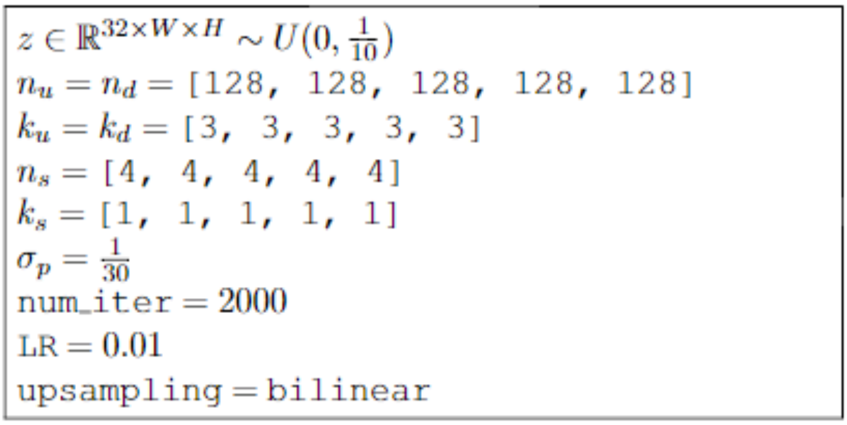
output image



bicubic

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

  - Network 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 $= 2000$

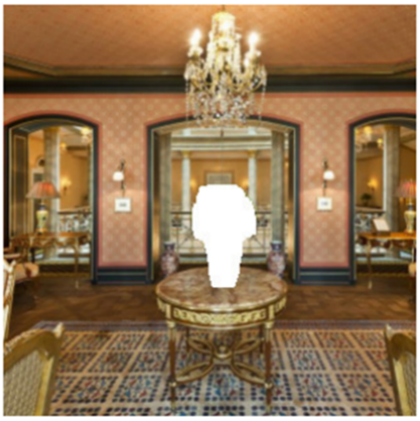
LR $= 0.01$

upsampling $=$ bilinear

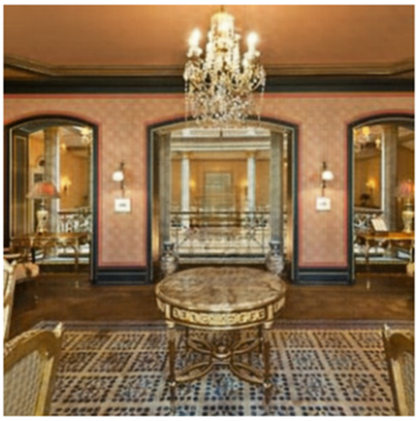
Optimizer: Adam

Extra Bonus (+2):

- Inpainting



(a) Corrupted image



(c) Ours, LR = 0.1

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

m: mask

References:

[1] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2017). Deep Image Prior. *arXiv preprint arXiv:1711.10925*.

- Report Spec: [black: Demo, Gray: No Demo]
- 1. Introduction (5%)
- 2. Experiment setup (5%)
  - The detail of your model
  - Report all your training hyper-parameters
- 3. Result
  - Requirement 1 (10%, 20%)
    - Training loss curve (you need to record training loss every iteration)
  - Requirement 2 (10%, 20%)
    - Visualize the progress of inverted image
    - Final image and its PSNR
  - Requirement 3 (10%, 20%)
    - Visualize the progress of inverted image
    - Final image and its PSNR
- 4. Discussion (10%, 20%)
- Demo (50%)
- **Demo 會給其他圖案，用你的程式重現實驗**

---- Criterion of result (denoising)(20%) ----

PSNR > 30 dB = 100%

PSNR 28.5~30 dB = 90%

PSNR 27~28.5 dB = 80%

PSNR < 27 dB = 70%

---- Criterion of result (super-resolution)(20%) ----

PSNR > 23 dB = 100%

PSNR 21.5~23 dB = 90%

PSNR 20~21.5 dB = 80%

PSNR < 20 dB = 70%

PSNR 計算請使用

from skimage.measure import compare_psnr

評分標準: 40%*實驗結果 + 60%*(報告+DEMO)