Learning optimized reaction diffusion processes for effective image restoration

Project #2

Tianfeng Lyu
Dylan Rachwal

Outline

- Context and proposed model of the paper
- Our benchmark
- Experiments

Context

- Image restoration with both high quality and high computational efficiency
- Nonlinear anisotropic diffusion (Perona-Malik diffusion process)
- learn PDEs from training data via an optimal control approach
- Pytorch unofficial implementation of the paper by Jean Plumail

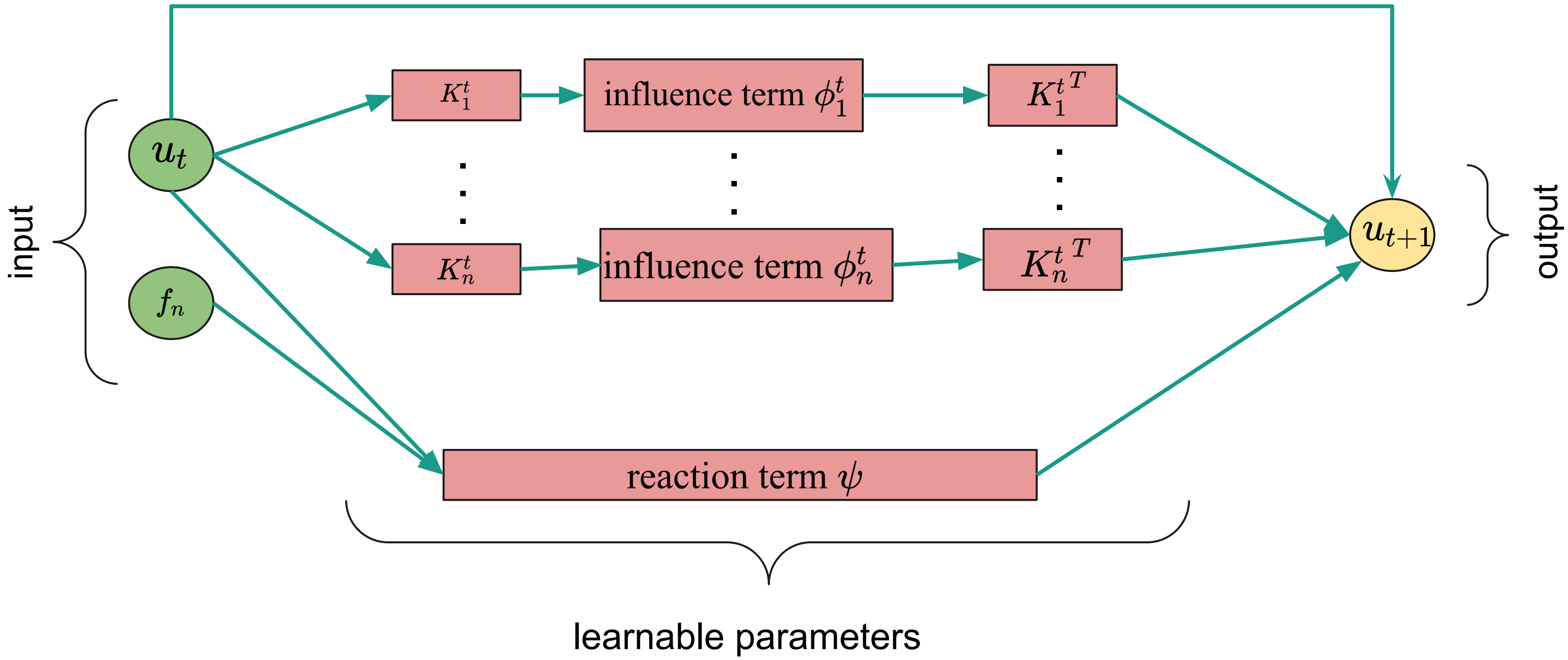
  jplumail/learning-image-restoration: Pytorch implementation of "On learning optimized reaction diffusion processes for effective image restoration" (github.com)

Proposed model

Learning model architecture

$$u_{t+1} = u_t - \Delta t\left(\sum_{i=1}^{N_k} {K_i^t}^T \phi_i^t(K_i^t u_t) - \psi(u_t, f_n)\right)$$

most used reaction term : $\psi(u, f_n) = \lambda A^T(Au - f_n)$



Benchmark

Benchmark

Different types of noise were used to benchmark the paper :

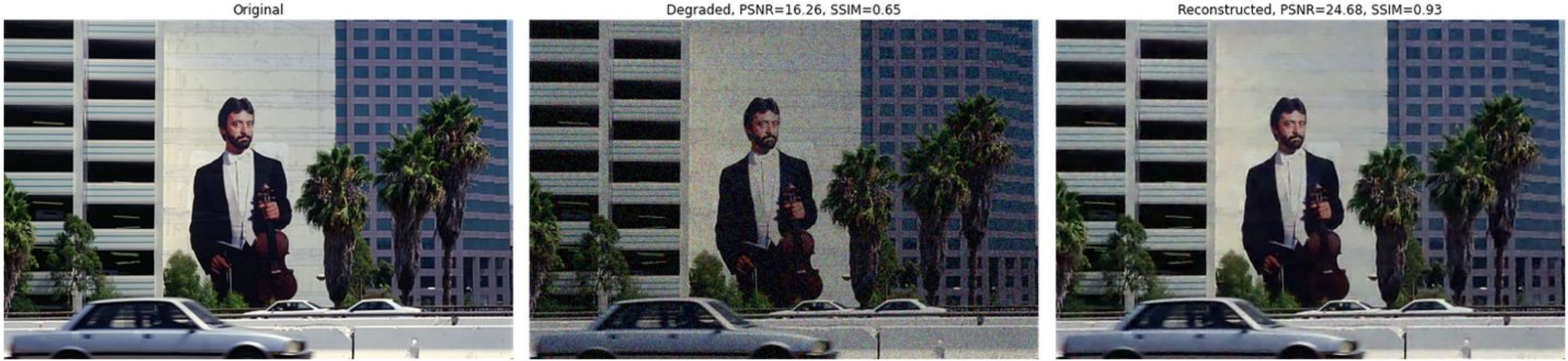
- Gaussian (already done in this paper)
- Salt and Pepper (2 versions, ours and skimage)
- Poisson (skimage)
- Speckle (skimage)

In that way, the only term changed in the model were the reaction term :

$$\psi(u, f_n) = \lambda(u - f_n) \quad \text{with} \quad f_n = x + \omega$$

Gaussian noise

$\psi(u, f_n) = \lambda(u - f_n)$ with $f_n = x + \omega$ and $\omega \sim \mathcal{N}(0, \sigma^2)$



Salt and pepper noise

$$\psi(u, f_n) = \lambda(u - f_n)$$

with

$$P(f_n(p) = x) = 1 - t$$
$$P(f_n(p) = min) = t/2$$
$$P(f_n(p) = max) = t/2$$

Salt and pepper noise

$$\psi(u, f_n) = \lambda(u - f_n)$$

with

$$P(f_n(p) = x) = 1 - t$$
$$P(f_n(p) = min) = t/2$$
$$P(f_n(p) = max) = t/2$$

Speckle noise

$$\psi(u, f_n) = \lambda(u - f_n) \quad \text{with} \quad f_n = x + x \times \omega \ \text{and} \quad \omega \sim \mathcal{N}(0, \sigma^2)$$

Original

Degraded, PSNR=17.01, SSIM=0.59

Reconstructed, PSNR=23.7, SSIM=0.9

Speckle or Gaussian ? $f_n = x + x \times \omega$ and $\omega \sim \mathcal{N}(0, \sigma^2)$

What happens if we
use Gaussian model ?

$$f_n = x + \omega$$

Original

Degraded, PSNR=17.0, SSIM=0.58

Reconstructed, PSNR=21.06, SSIM=0.8

Original

Degraded, PSNR=17.07, SSIM=0.65

Reconstructed, PSNR=22.16, SSIM=0.89

Poisson Noise

$$\psi(u, f_n) = \lambda(u - f_n) \quad \text{with} \quad f_n(p) \sim P(x(p))$$

Appendix

Training scheme used :

- Filter size = 5 x 5 / 7x7
- Number of filters = 24 / 48
- Greedy training (finetuned afterwards with joint training)

50 epochs (approx 1 hour) with early stop

adaptative learning rate