IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Learning Image restoration using optimized reaction diffusion processes.

- Badr SOULAIMANI
- Ilias ELFRYAKH

UE CompImaging
21/03/2023

Citation

This presentation is based on the work of :

Yunjin Chen, Wei Yu, Thomas Pock

Entitled

On learning optimized reaction diffusion processes for effective image restoration

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

PLAN

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

PROBLEMATIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Image restoration

The process of estimation uncorrupted images from noisy or blurred ones.

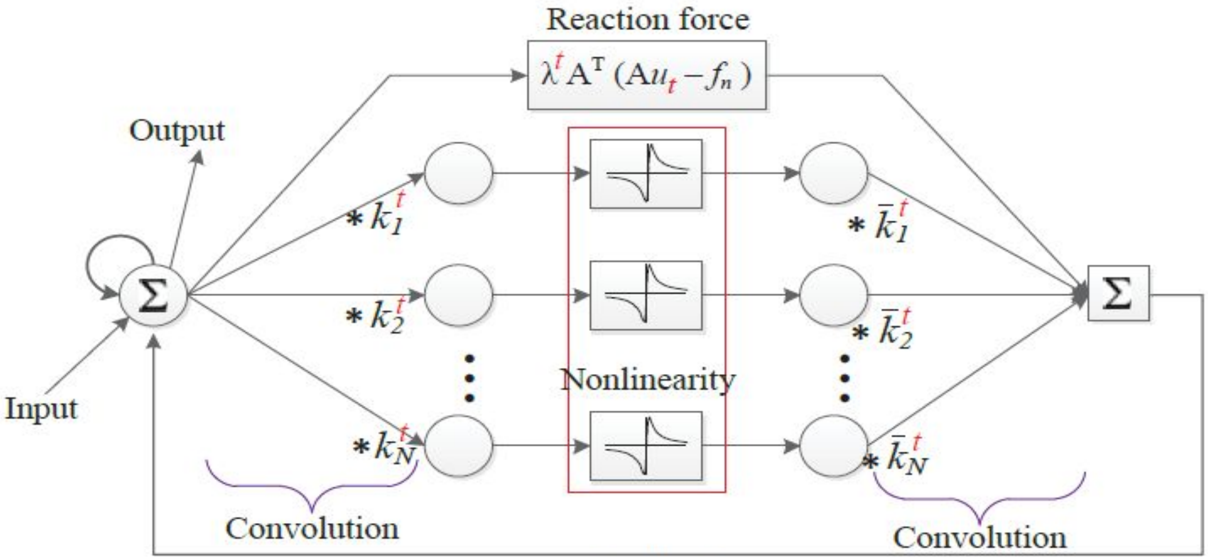
High quality restoration
vs
High computational efficiency

METHODOLOGY

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Model

The architecture of the model used (RNN = CNN with feedback)



Learning

RD Equation

$$\frac{u_t - u_{t-1}}{\Delta t} = -\underbrace{\sum_{i=1}^{N_k} K_i^{t\top} \phi_i^t(K_i^t u_{t-1})}_{\text{diffusion term}} - \underbrace{\psi(u_{t-1}, f_n)}_{\text{reaction term}}$$

$\xrightarrow{\text{Denoising}}$

$$u_t = u_{t-1} - \left(\sum_{i=1}^{N_k} \bar{k}_i^t * \phi_i^t(k_i^t * u_{t-1}) + \lambda^t (u_{t-1} - f_n) \right)$$

Influence function

$$\phi_i^t(z) = \sum_{j=1}^{M} w_{ij}^t \varphi\left(\frac{|z - \mu_j|}{\gamma_j}\right)$$

Gaussian RBFs

$$\varphi_g(z) = \varphi\left(\frac{|z-\mu|}{\gamma}\right) = \exp\left(-\frac{(z-\mu)^2}{2\gamma^2}\right)$$

Learning

Loss functions

- Joint training: $$\mathcal{L}(\Theta_{1,\cdots,T}) = \sum_{s=1}^{S} \ell(u_T^{(s)}, u_{gt}^{(s)})$$
- Greedy training: $$\mathcal{L}(\Theta_t) = \sum_{s=1}^{S} \ell(u_t^{(s)}, u_{gt}^{(s)})$$

MSE

$$\ell(u_t^{(s)}, u_{gt}^{(s)}) = \frac{1}{2}\|u_t^{(s)} - u_{gt}^{(s)}\|_2^2$$

With : $\Theta_t = \{\lambda^t, \phi_i^t, k_i^t\}$

Evaluation metric

$$\mathrm{PSNR}(I, J) = 10 * \log_{10}\left(\frac{\max(I)^2}{\mathrm{MSE}(I, J)}\right)$$

IMPLEMENTATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2. IMPLEMENTATION

Model

- Pytorch implementation
- TRD model with 8 stages, 48 filters of size 7x7
- 63 Gaussian RBFs same as the author’s implementation

Training

- Image dataset(same as the authors, 300 images) with additive gaussian noise(σ=25)\Poisson noise.
- Training with greedy or joint mode ⇒ Possibility of training with greedy mode then fine tune the model with joint mode.
- An Adam optimizer was used instead of the L-BFGS algorithm. During training, 180x180 patches randomly cropped from training images are fed to the network.
- The learning phase uses backpropagation from Pytorch on the GPU (VM) whereas authors used explicit derivatives on CPU in Matlab.

EXPERIMENTS

2. Comparison to Other Methods

Method	σ 15	σ 25	St.	$\sigma = 15$ TRD$_{5\times5}$	$\sigma = 15$ TRD$_{7\times7}$
BM3D	31.08	28.56	2	31.14	31.30
LSSC	31.27	28.70	5	31.30	**31.42**
EPLL	31.19	28.68	8	31.34	**31.43**
opt-MRF	31.18	28.66		$\sigma = 25$	
RTF_5	–	28.75		TRD$_{5\times5}$	TRD$_{7\times7}$
WNNM	31.37	28.83	2	28.58	28.77
$CSF^5_{5\times5}$	31.14	28.60	5	28.78	**28.92**
$CSF^5_{7\times7}$	31.24	28.72	8	28.83	**28.95**

Table 1. Average PSNR (dB) on 68 images from [36] for image denoising with $\sigma = 15, 25$.

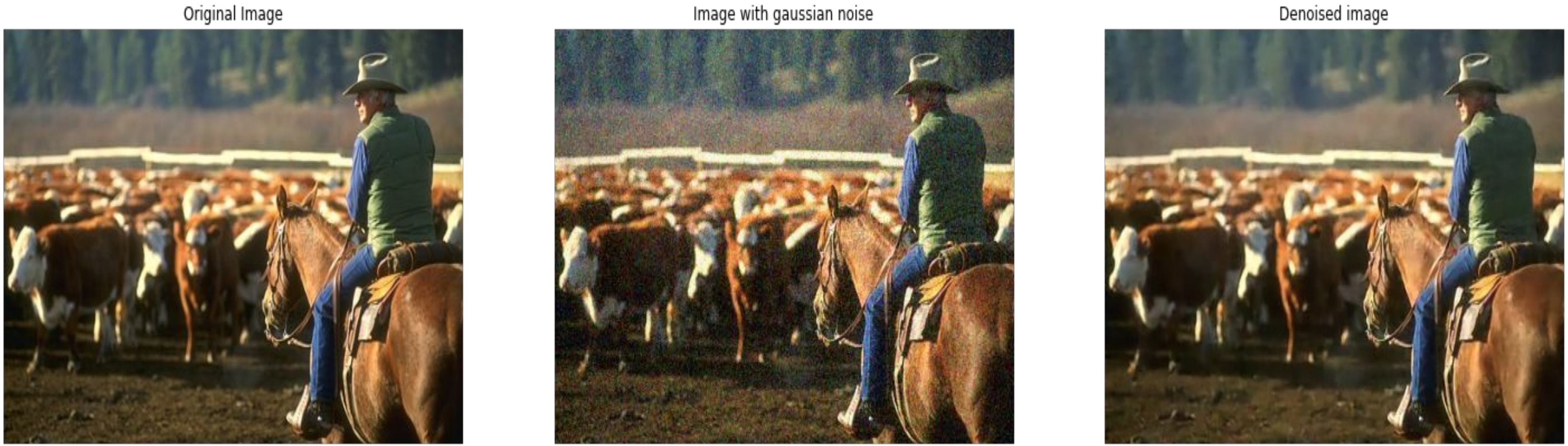
Method	256^2	512^2	1024^2	2048^2	3072^2
BM3D [11]	1.1	4.0	17	76.4	176.0
$CSF^5_{7\times7}$ [38]	3.27	11.6	40.82	151.2	494.8
WNNM [19]	122.9	532.9	2094.6	–	–
$TRD^5_{5\times5}$	0.51	1.53	5.48	24.97	53.3
	0.43	0.78	2.25	8.01	21.6
	0.005	0.015	0.054	0.18	0.39
$TRD^5_{7\times7}$	1.21	3.72	14.0	62.2	135.9
	0.56	1.17	3.64	13.01	30.1
	0.01	0.032	0.116	0.40	0.87

Table 2. Run time comparison for image denoising (in seconds) with different implementations. (1) The run time results with gray background are evaluated with the single-threaded implementation on Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz; (2) the blue colored run times are obtained with multi-threaded computation using Matlab *parfor* on the above CPUs; (3) the run time results colored in red are executed on a NVIDIA GeForce GTX 780Ti GPU. We do not count the memory transfer time between CPU/GPU for the GPU implementation (if counted, the run time will nearly double)

2. OWN EXPERIMENTS

PSNR (dB)									
	Sigma = 15			Sigma = 25			Poisson		
Stage	TRD 7x7 (Greedy+Joint Gaussian)	TRD 7x7 (Joint Gaussian)	TRD 7x7 (Joint Poisson)	TRD 7x7 (Greedy+Joint Gaussian)	TRD 7x7 (Joint Gaussian)	TRD 7x7 (Joint Poisson)	TRD 7x7 (Greedy+Joint Gaussian)	TRD 7x7 (Joint Gaussian)	TRD 7x7 (Joint Poisson)
2	27.56	5.91	2.28	24.96	5.76	2.13	28.01	5.95	2.33
5	30.02	-9.44	-11.55	27.72	-9.45	-11.55	30.33	-9.44	-11.54
8	31.02	28.66	30.8	30.41	27.15	26.01	30.96	29.01	33.84

2. SAMPLES



CONCLUSION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Take Away Messages

- Transform an already used method (Reaction diffusion equation) to a trainable model and optimize it using a differentiable framework.

- Visualizing the learned kernels and functions provides a level of interpretability to the model.

- Wide range of applications to this kind of approach.