Lecture 3 Image Denoising

Yuyao Zhang PhD
zhangyy8@shanghaitech.edu.cn
SIST Building-3 420

上海科技大学
ShanghaiTech University

Outline

- Conventional denoising
  - Total Variation denoising
  - Markov Random Fields
  - Dictionary-learning

Resource：https://github.com/wenbihan/reproducible-image-denoising-state-of-the-art

Total Variation Denoising

[1] L. Rudin and S. Osher, “Total variation based image restoration with free local constraints,” in Proc. 1st IEEE Int. Conf. Image Processing, vol. 1, 1994, pp. 31–35
[2] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Phys. D, vol. 60, pp. 259–268, 1992.

Total Variation

- The intuitive idea of ***Total Variation*** is based on the premise that signals exhibiting excessive and potentially spurious details possess high total variation (the integral of the image gradient magnitude).
- For a digital signal x_n, we can, for example, define the total variation as

$$V(x) = \sum_n |x_{n+1} - x_n|$$

- To recover the clean signal x_n from y_n is defined by total-variation denoising problem amounts to minimizing the following discrete functional over the signal :

$$E(x, y) + \lambda V(x) = \frac{1}{n} \sum_n (x_n - y_n)^2 + \lambda * \sum_n |x_{n+1} - x_n|$$

- In the original approach, this function is derived using **Euler–Lagrange equation**. Since it is a convex function, techniques from **convex optimization** can be used to minimize it and find the solution.

Total Variation

- We now consider 2D signals x, such as images. The total-variation norm proposed by the 1992 article is

$$V(x) = \sum_{i,j} \sqrt{\left|x_{i+1,j} - x_{i,j}\right|^2 + \left|x_{i,j+1} - x_{i,j}\right|^2}$$

- It is isotropic and not differentiable. A variation that is sometimes used, since it may sometimes be easier to minimize, is an anisotropic version

$$V_{aniso}(x) = \left|x_{i+1,j} - x_{i,j}\right| + \left|x_{i,j+1} - x_{i,j}\right|$$

- The standard total-variation denoising problem is still of the form

$$\min_x [E(x, y) + \lambda V(x)]$$

- where $E(x, y)$ is the 2D L2 norm. In contrast to the 1D case, solving this denoising is non-trivial. The authors in [1] solved this by s the primal dual method [2].

[1] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Phys. D, vol. 60, pp. 259–268, 1992.
[2] Chambolle, A. (2004). "An algorithm for total variation minimization and applications". Journal of Mathematical Imaging and Vision. **20**: 89–97.

Marcov Random Fields

[1] Boykov, Y. and Kolmogorov, V. (2004) An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans PAMI 26(9):1124-1137.
[2] Boykov, Y., Veksler, O., and Zabih, R. (2001) Fast approximate energy minimization via graph cuts. IEEE Trans. PAMI, 23(11):1222-1239.

Dictionary-learning Denoising

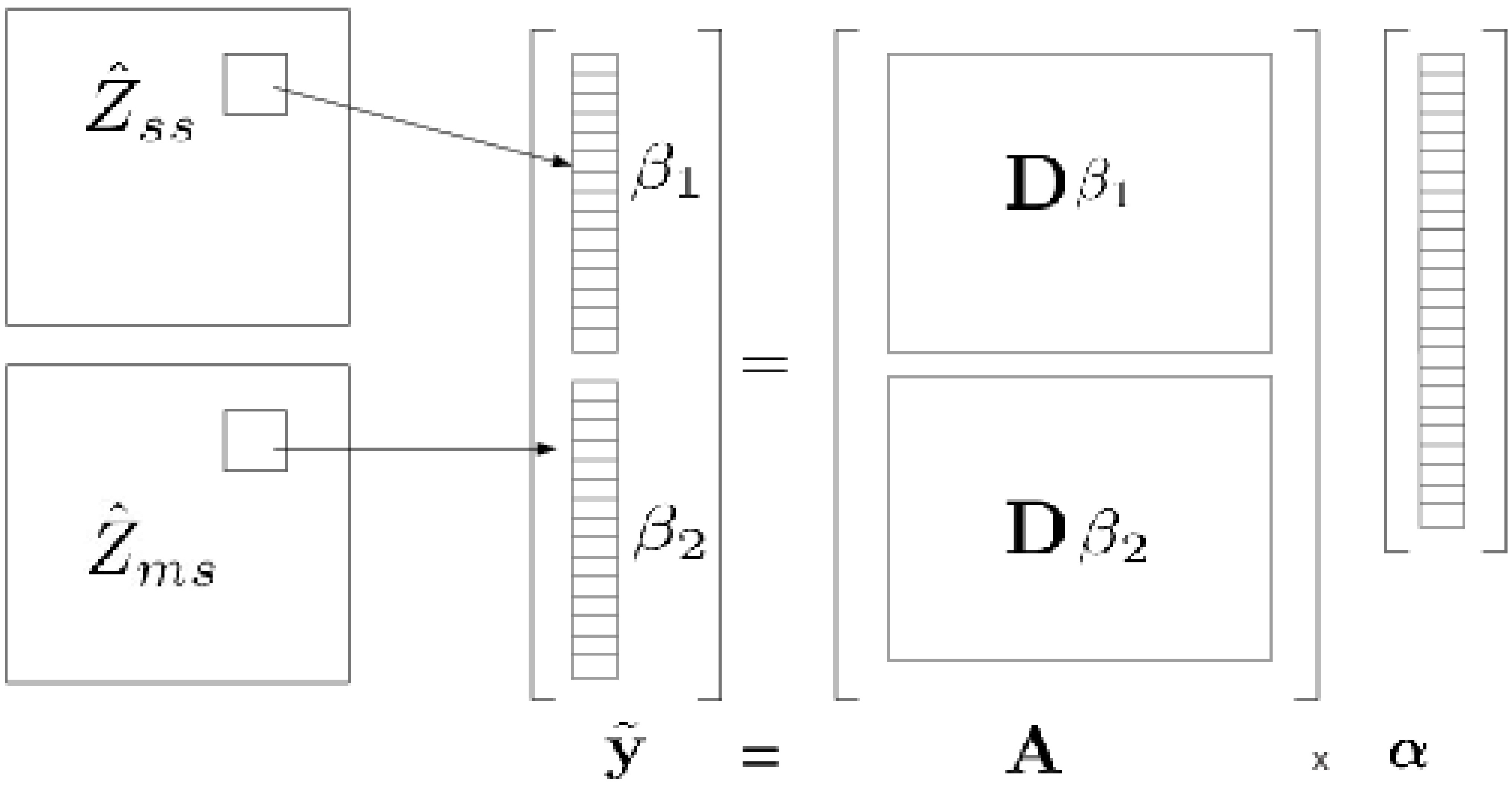
reference :
[1] C. Taswell, "The what, how, and why of wavelet shrinkage denoising," in Computing in Science & Engineering, vol. 2, no. 3, pp. 12-19, May-June 2000, doi: 10.1109/5992.841791.
[2] Meyer Scetbon; Michael Elad; Peyman Milanfar, et al. Deep K-SVD Denoising. IEEE Transactions on image processing, 2021, 16(8): 5944-5955.
[3] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. on Signal Processing, 54(11):4311, 2006

Overview

- The basic idea behind wavelet denoising, or any linear transform thresholding, is that the transform leads to a sparse representation for many real-world signals and images.
- What this means is that the wavelet transform concentrates signal and image features in a few large-magnitude wavelet coefficients.
- Wavelet coefficients which are small in value are typically noise and you can "shrink" those coefficients or remove them without affecting the signal or image quality.
- After you threshold the coefficients, you reconstruct the data using the inverse transform

Dictionary Selection

- What D to use?
- A fixed overcomplete set of basis:
  - Steerable wavelet
  - Contourlet
  - DCT Basis
- Data Adaptive Dictionary – learn from data


$\hat{Z}_{ss}$
$\hat{Z}_{ms}$
β_1
β_2
$=$
$\mathbf{D}_{\beta_1}$
$\mathbf{D}_{\beta_2}$
$\tilde{\mathbf{y}}$
$=$
$\mathbf{A}$
x
$\boldsymbol{\alpha}$

Denoising Framework

- A cost function for: $\boldsymbol{Z = Y + n}$

$$\boldsymbol{arg\ min_Y \|Y - Z\|_2^2 + \lambda\|\cdot\|_p}$$

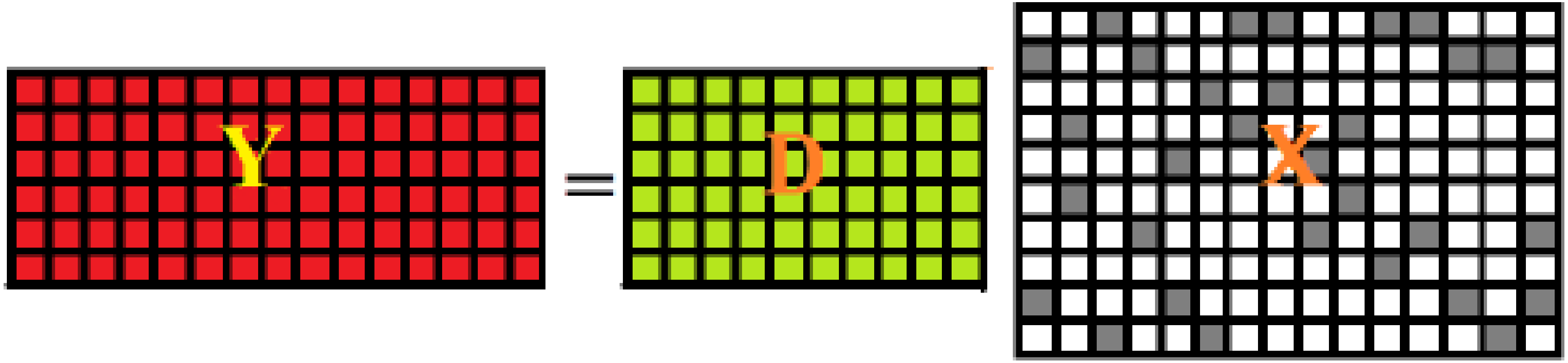
Prior Term

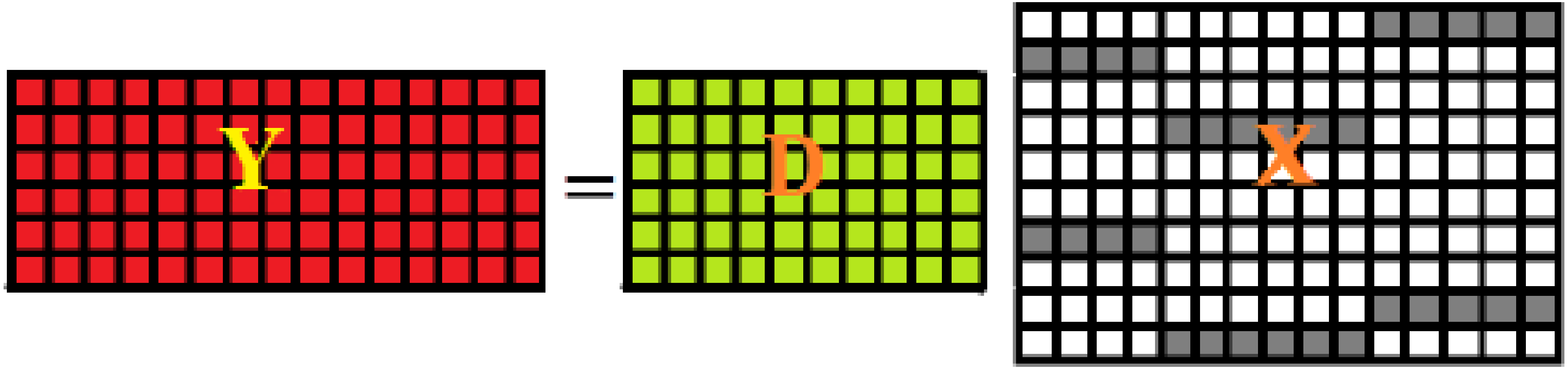
- Solve for

$$\boldsymbol{\hat{X}, \hat{Y} = arg\ min_{X,Y} \|Y - Z\|_2^2 + \lambda\|DX - Y\|_F^2 + \sum_i \mu_i \|x_i\|_0}$$

Dictionary Learning

Sparse Representation


Y
=
D
X


Y
=
D
X

Supervised Deep Learning Approaches

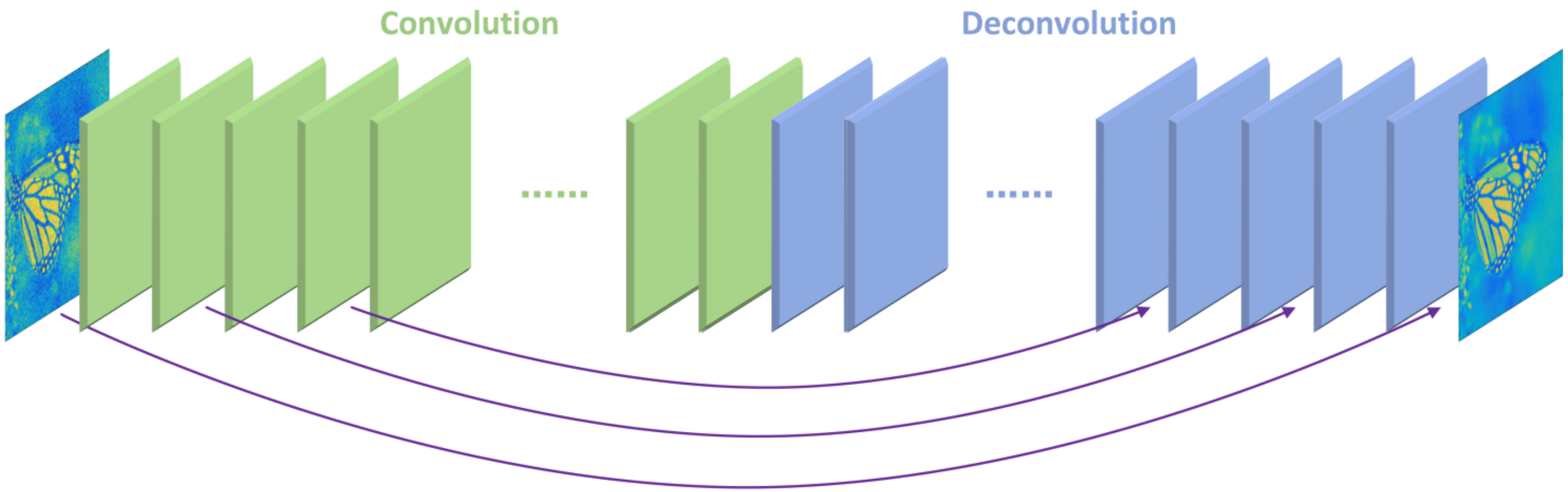
RED： Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections

[1] Xiao-Jiao Mao , Chunhua Shen, Yu-Bin Yang： Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections (NIPS2016), Mao et al. 2016

RED – Key Contribution

- The network is composed of multiple layers of convolution and de-convolution operators, learning end-to-end mappings from corrupted images to the original ones.
- We propose to symmetrically link convolutional and de-convolutional layers with skip-layer connections, with which the training converges much faster and attains a higher-quality local optimum.
- Significantly, with the large capacity, we can handle different levels of noises using a single model.

A deep noise to clean denoising method



We minimize the following Mean Squared Error(MSE):

$$\mathcal{L}(\Theta) = \frac{1}{n} \sum_{n=1}^{N} \|\mathcal{F}(X_i; \Theta) - Y_i\|_F^2. \tag{1}$$

Quantitative Evaluation Parameter

Table 1: Average PSNR and SSIM results of σ 10, 30, 50, 70 for the 14 images.

	PSNR								
	BM3D	EPLL	NCSR	PCLR	PGPD	WNNM	RED10	RED20	RED30
$\sigma = 10$	34.18	33.98	34.27	34.48	34.22	34.49	34.62	34.74	**34.81**
$\sigma = 30$	28.49	28.35	28.44	28.68	28.55	28.74	28.95	29.10	**29.17**
$\sigma = 50$	26.08	25.97	25.93	26.29	26.19	26.32	26.51	26.72	**26.81**
$\sigma = 70$	24.65	24.47	24.36	24.79	24.71	24.80	24.97	25.23	**25.31**
	SSIM								
$\sigma = 10$	0.9339	0.9332	0.9342	0.9366	0.9309	0.9363	0.9374	0.9392	**0.9402**
$\sigma = 30$	0.8204	0.8200	0.8203	0.8263	0.8199	0.8273	0.8327	0.8396	**0.8423**
$\sigma = 50$	0.7427	0.7354	0.7415	0.7538	0.7442	0.7517	0.7571	0.7689	**0.7733**
$\sigma = 70$	0.6882	0.6712	0.6871	0.6997	0.6913	0.6975	0.7012	0.7177	**0.7206**

Quantitative Evaluation Parameter

- PSNR (Peak signal to noise ratio)

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

Quantitative Evaluation Parameter

- SSIM (Structure Similarity Index Measure)

  - Luminance

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

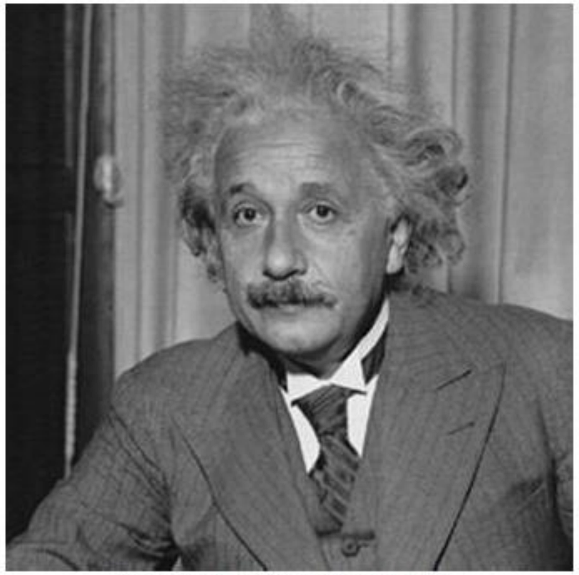
  - Contrast

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}} \qquad l(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

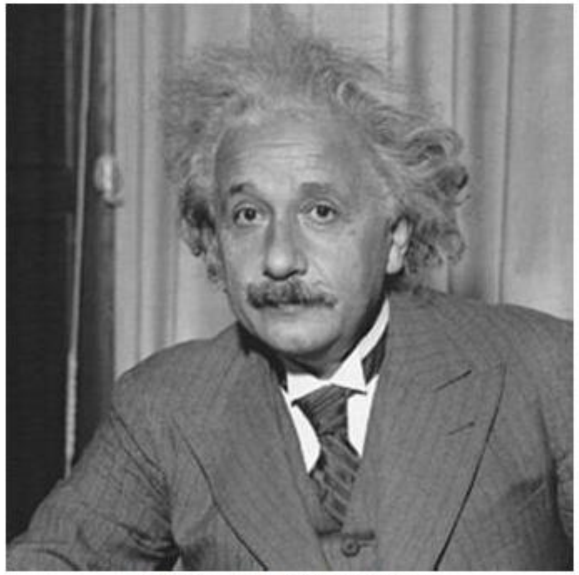
  - Structure

$$c(x,y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \qquad \sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

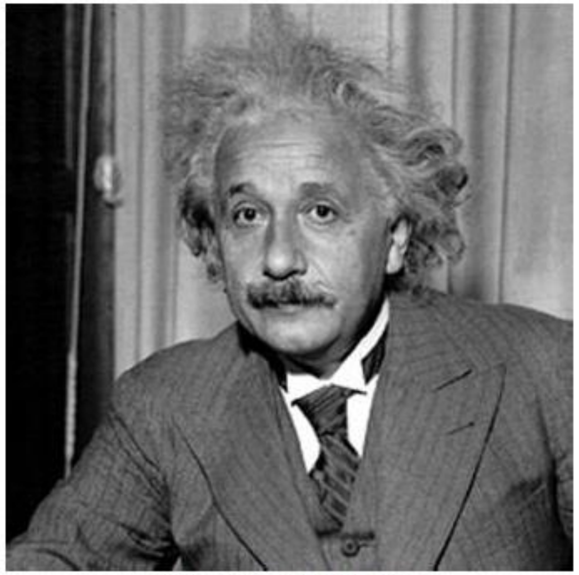
$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$



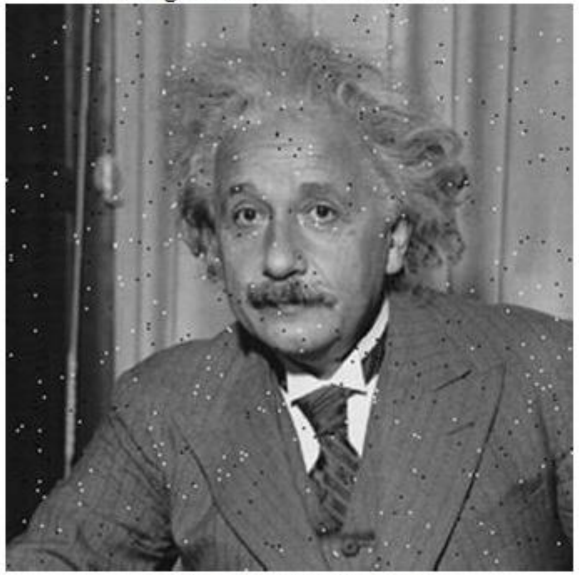
Original, MSE = 0; SSIM = 1



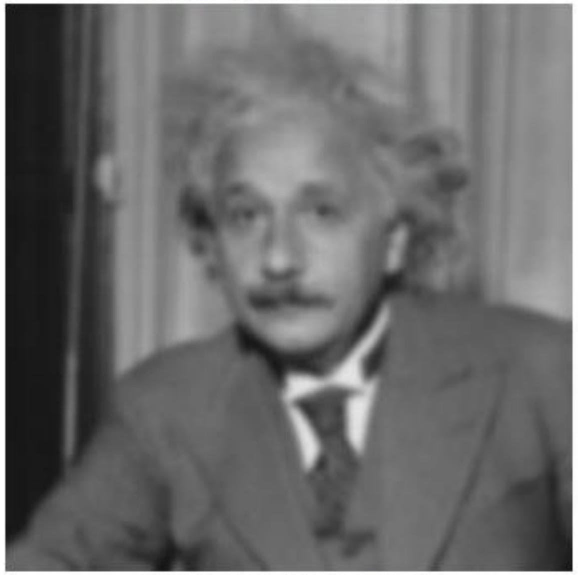
MSE = 144, SSIM = 0.988



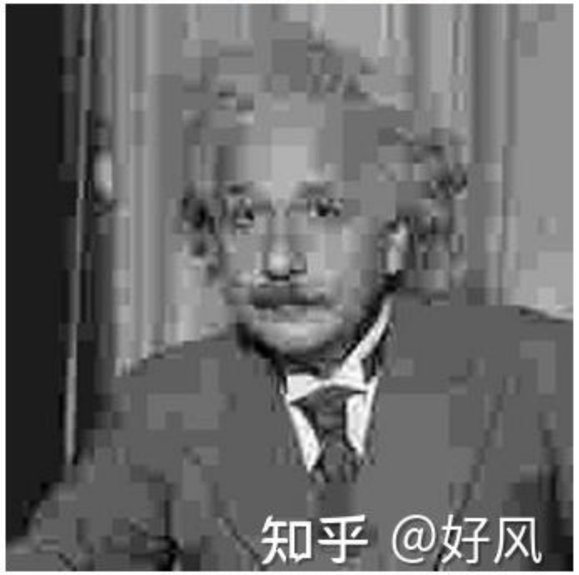
MSE = 144, SSIM = 0.913



MSE = 144, SSIM = 0.840



MSE = 144, SSIM = 0.694



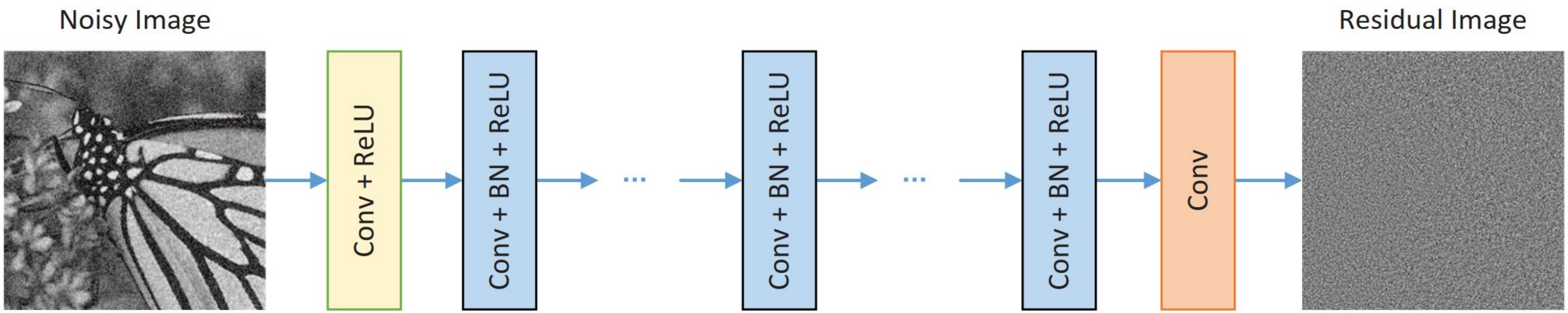
MSE = 142, SSIM = 0.662

DnCNN: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising

[1] FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (TIP 2018), Zhang et al.

DnCNN– Key Contribution

- With the residual learning strategy, DnCNN implicitly removes the latent clean image in the hidden layers.
- Specifically, residual learning and batch normalization are utilized to speed up the training process as well as boost the denoising performance.



Deep Residual Learning for Image Recognition

- Motivation

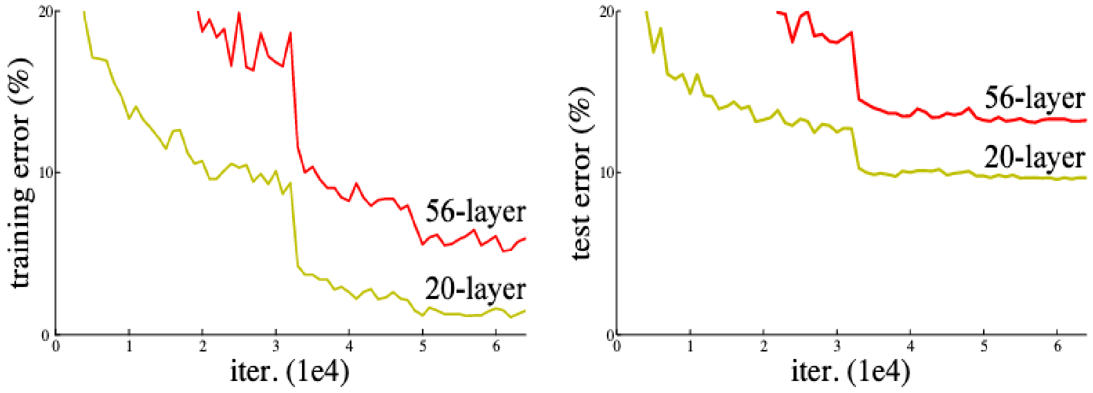


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- Structure

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

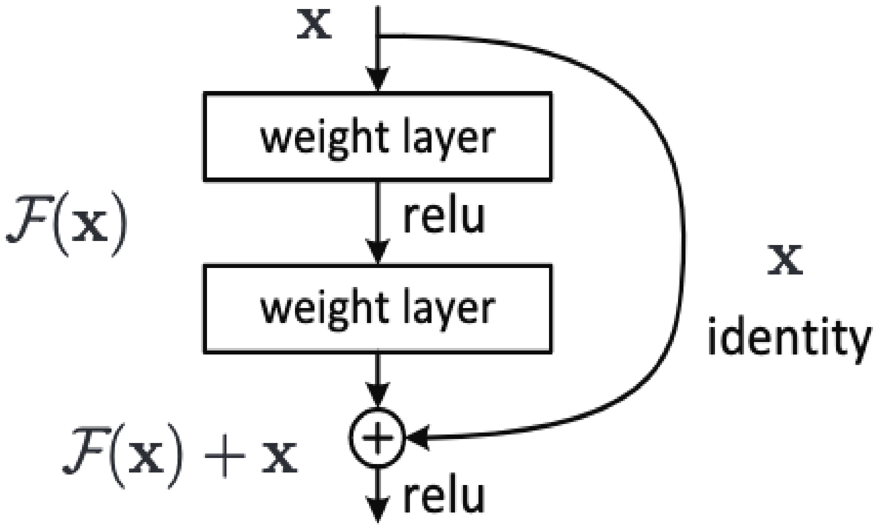
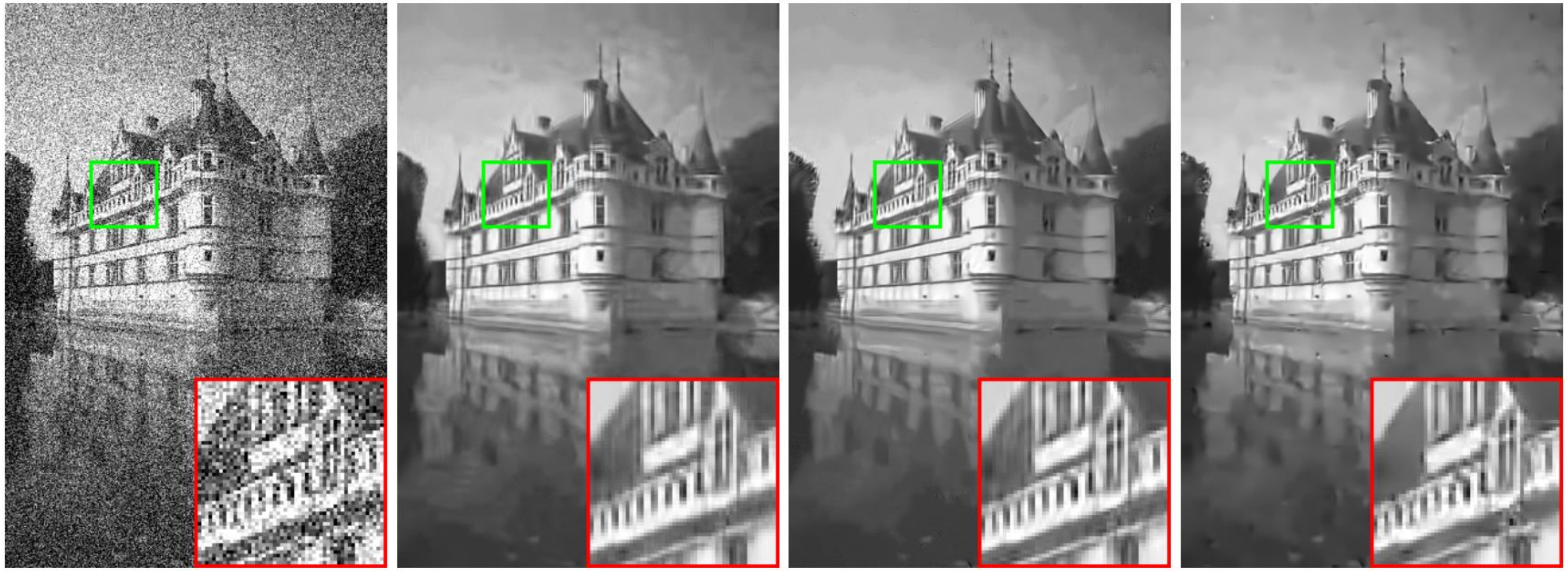
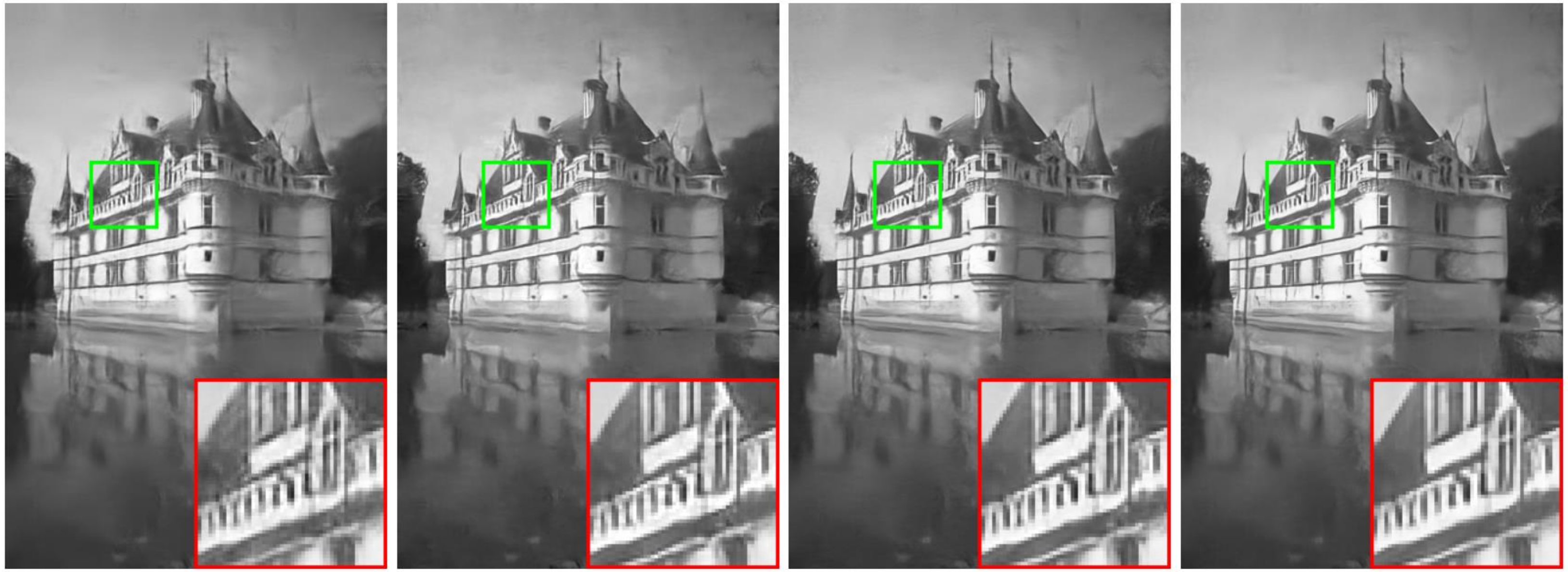


Figure 2. Residual learning: a building block.

上海科技大学
ShanghaiTech University



(a) Noisy / 14.76dB (b) BM3D / 26.21dB (c) WNNM / 26.51dB (d) EPLL / 26.36dB



(e) MLP / 26.54dB (f) TNRD / 26.59dB (g) DnCNN-S / 26.90dB (h) DnCNN-B / 26.92dB

Fig. 4. Denoising results of one image from BSD68 with noise level 50.

FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising

[1] FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (TIP 2018), Zhang et al.

FFDNet – Key Contribution

- A fast and flexible denoising network, namely FFDNet, is proposed for discriminative image denoising. By taking a tunable noise level map as input, a single FFDNet is able to deal with noise on different levels, as well as spatially variant noise.
- We highlight the importance to guarantee the role of the noise level map in controlling the trade-off between noise reduction and detail preservation.
- FFDNet exhibits perceptually appealing results on both synthetic noisy images corrupted by AWGN and real world noisy images, demonstrating its potential for practical image denoising.

Pipeline

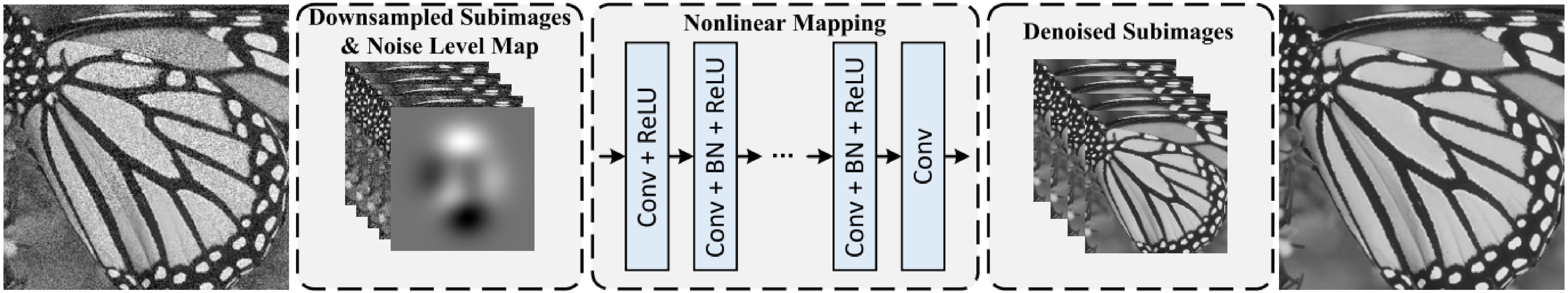


Fig. 1. The architecture of the proposed FFDNet for image denoising. The input image is reshaped to four sub-images, which are then input to the CNN together with a noise level map. The final output is reconstructed by the four denoised sub-images.

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + \lambda\boldsymbol{\Phi}(\boldsymbol{x}), \tag{1}$$

where $\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2$ is the data fidelity term with noise level σ, $\boldsymbol{\Phi}(\boldsymbol{x})$ is the regularization term associated with image prior, and λ controls the balance between the data fidelity and regularization terms. It is worth noting that in practice λ governs the compromise between noise reduction and detail preservation. When it is too small, much noise will remain; on the opposite, details will be smoothed out along with suppressing noise.

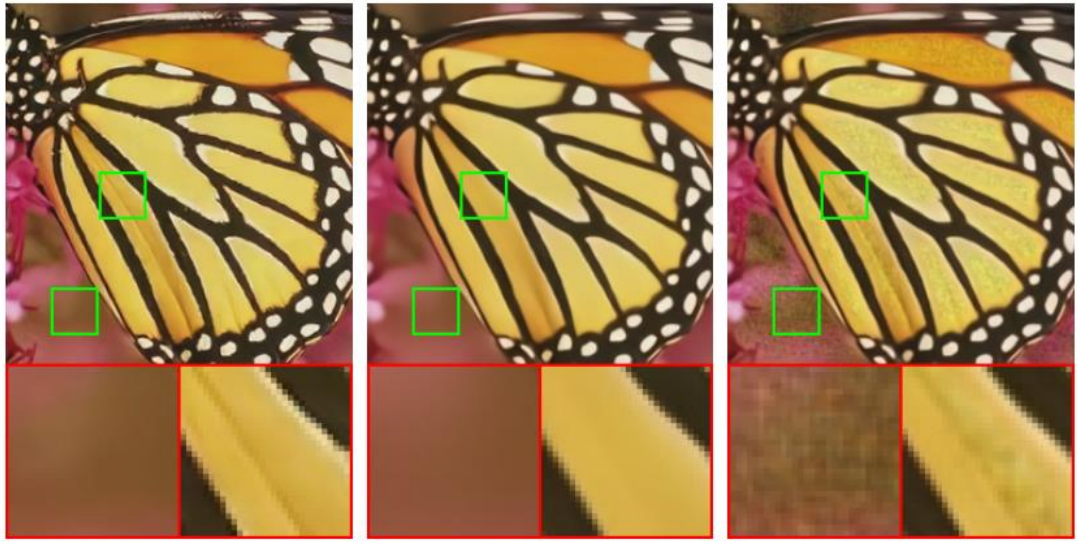
With some optimization algorithms, the solution of Eqn. (1) actually defines an implicit function given by

$$\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{y}, \sigma, \lambda; \Theta). \tag{2}$$

Since λ can be absorbed into σ, Eqn. (2) can be rewritten as

$$\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{y}, \sigma; \Theta). \tag{3}$$

Pipeline



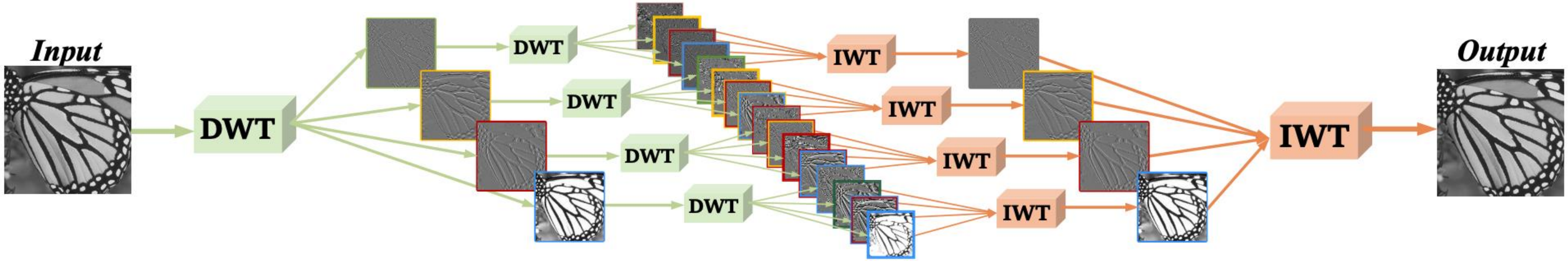
(a) (b) (c)

Fig. 2. An example to show the importance of guaranteeing the role of noise level map in controlling the trade-off between noise reduction and detail preservation. The input is a noisy image with noise level 25. (a) Result without visual artifacts by matched noise level 25. (b) Result without visual artifacts by mismatched noise level 60. (c) Result with visual artifacts by mismatched noise level 60.

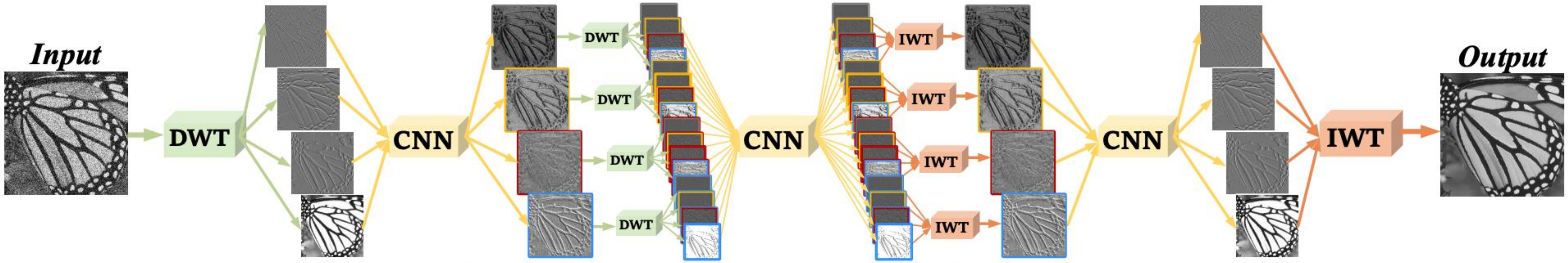
Images	*C.man*	*House*	*Peppers*	*Starfish*	*Monarch*	*Airplane*	*Parrot*	*Lena*	*Barbara*	*Boat*	*Man*	*Couple*	*Average*
Noise Level	$\sigma = 15$												
BM3D	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.37
WNNM	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.70
TNRD	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.50
DnCNN	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.86
FFDNet	32.42	35.01	33.10	32.02	32.77	31.58	31.77	34.63	32.50	32.35	32.40	32.45	32.75
Noise Level	$\sigma = 25$												
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.97
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.26
MLP	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.03
TNRD	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.06
DnCNN	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.43
FFDNet	30.06	33.27	30.79	29.33	30.14	29.05	29.43	32.59	29.98	30.23	30.10	30.18	30.43
Noise Level	$\sigma = 35$												
BM3D	27.92	31.36	28.51	26.86	27.58	26.83	27.40	30.56	28.98	28.43	28.22	28.15	28.40
WNNM	28.08	31.92	28.75	27.27	28.13	27.10	27.69	30.73	29.48	28.54	28.33	28.24	28.69
MLP	28.08	31.18	28.54	27.12	27.97	27.22	27.72	30.82	27.62	28.53	28.47	28.24	28.46
DnCNN	28.61	31.61	29.14	27.53	28.51	27.52	27.94	30.91	28.09	28.72	28.66	28.52	28.82
FFDNet	28.54	31.99	29.18	27.58	28.54	27.47	28.02	31.20	28.29	28.82	28.70	28.68	28.92
Noise Level	$\sigma = 50$												
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.72
WNNM	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.05
MLP	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.78
TNRD	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.81
DnCNN	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.18
FFDNet	27.03	30.43	27.43	25.77	26.88	25.90	26.58	29.68	26.48	27.32	27.30	27.07	27.32
Noise Level	$\sigma = 75$												
BM3D	24.32	27.51	24.73	23.27	23.91	23.48	24.18	27.25	25.12	25.12	25.32	24.70	24.91
WNNM	24.60	28.24	24.96	23.49	24.31	23.74	24.43	27.54	25.81	25.29	25.42	24.86	25.23
MLP	24.63	27.78	24.88	23.57	24.40	23.87	24.55	27.68	23.39	25.44	25.59	25.02	25.07
DnCNN	25.07	27.85	25.17	23.64	24.71	24.03	24.71	27.54	23.63	25.47	25.64	24.97	25.20
FFDNet	25.29	28.43	25.39	23.82	24.99	24.18	24.94	27.97	24.24	25.64	25.75	25.29	25.49

Multi-level Wavelet-CNN for Image Restoration

[1] Multi-level Wavelet-CNN for Image Restoration (CVPR 2018), Liu et al.



(a) Multi-level WPT architecture



(b) Multi-level wavelet-CNN architecture

Figure 2. From WPT to MWCNN. Intuitively, WPT can be seen as a special case of our MWCNN without CNN blocks.

Key Contribution

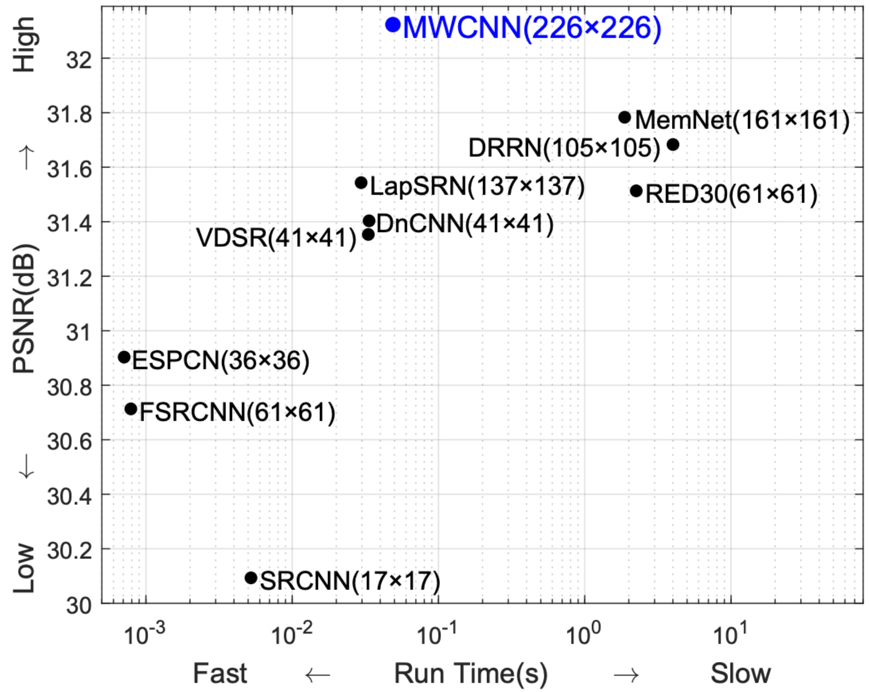


Figure 1. The run time vs. PSNR value of representative CNN models, including SRCNN [16], FSRCNN [14], ESPCN [45], VDSR [29], DnCNN [57], RED30 [37], LapSRN [31], DRRN [47], MemNet [47] and our MWCNN. The receptive field of each model are also provided. The PSNR and time are evaluated on Set5 with the scale factor ×4 running on a GTX1080 GPU.

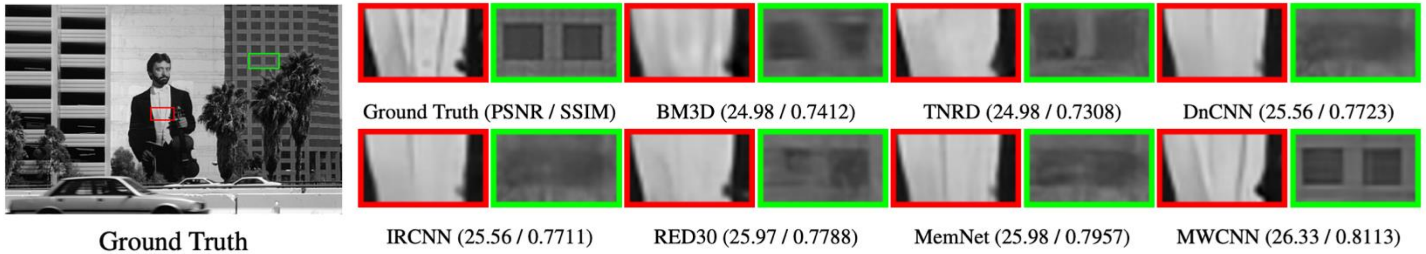


Figure 5. *Image denoising* results of "*Test*011" (BSD68) with noise level 50.

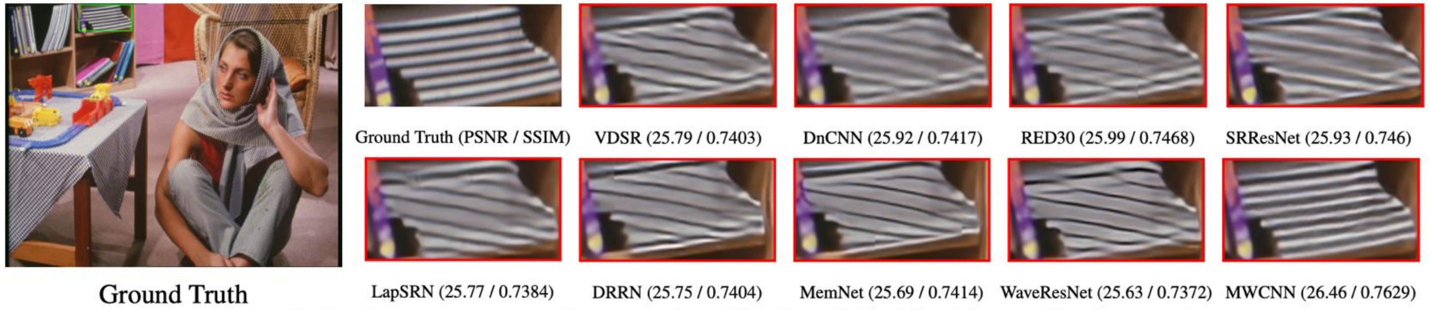


Figure 6. *Single image super-resolution* results of "*barbara*" (Set14) with upscaling factor ×4.

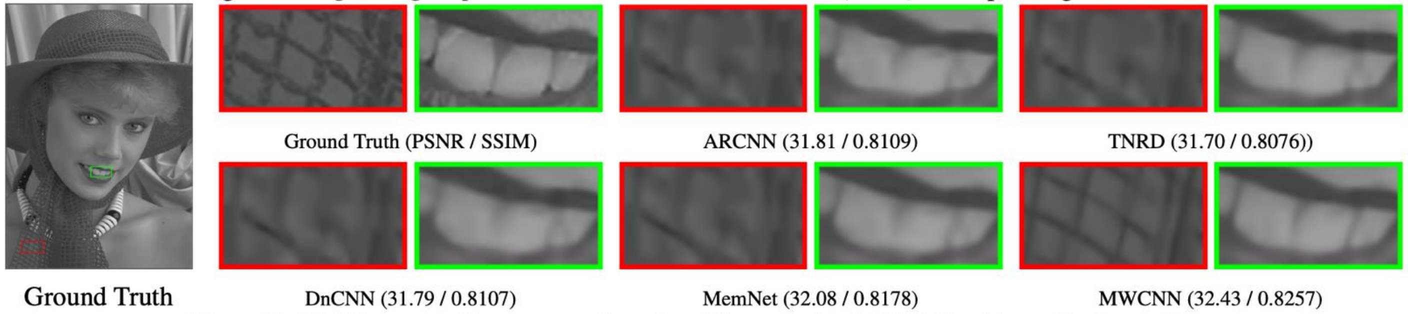


Figure 7. *JPEG image artifacts removal* results of "*womanhat*" (LIVE1) with quality factor 10.

上海科技大学 ShanghaiTech University

Image Denoising					
Size	TNRD [10]	DnCNN [57]	RED30 [37]	MemNet [47]	MWCNN
256×256	0.010	0.0143	1.362	0.8775	0.0586
512×512	0.032	0.0487	4.702	3.606	0.0907
1024×1024	0.116	0.1688	15.77	14.69	0.3575

Single Image Super-Resolution					
Size	VDSR [29]	LapSRN [31]	DRRN [47]	MemNet [37]	MWCNN
256×256	0.0172	0.0229	3.063	0.8774	0.0424
512×512	0.0575	0.0357	8.050	3.605	0.0780
1024×1024	0.2126	0.1411	25.23	14.69	0.3167

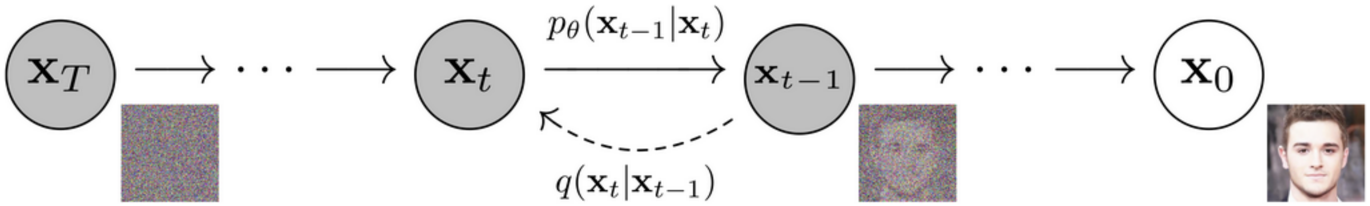
JPEG Image Artifacts Removal					
Size	ARCNN [15]	TNRD [10]	DnCNN [57]	MemNet [37]	MWCNN
256×256	0.0277	0.009	0.0157	0.8775	0.0531
512×512	0.0532	0.028	0.0568	3.607	0.0811
1024×1024	0.1613	0.095	0.2012	14.69	0.2931

Denoising Diffusion Probabilistic Models

[1] Denoising Diffusion Probabilistic Models (NeurIPS 2020), Ho et al.

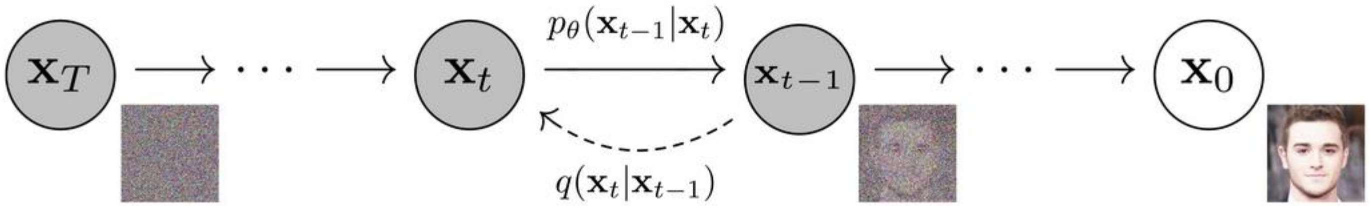
DDPM – Key Contribution

- High quality image synthesis with diffusion probabilistic models. Unconditional CIFAR10 FID=3.17, LSUN samples comparable to GANs.
- We show connections to denoising score matching + Langevin dynamics, yet we provide log likelihoods and rate-distortion curves.
- We demonstrate compression with controllable lossiness, allowing reconstructions and interpolations at multiple granularities.



Diffusion probabilistic models are parameterized Markov chains trained to gradually denoise data. We estimate parameters of the generative process p.

DDPM – Diffusion models



Diffusion models [53] are latent variable models of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latents of the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The joint distribution $p_\theta(\mathbf{x}_{0:T})$ is called the *reverse process*, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{1}$$

What distinguishes diffusion models from other types of latent variable models is that the approximate posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, called the *forward process* or *diffusion process*, is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2}$$

DDPM – Diffusion models

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
  $$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$$
end for
return $\mathbf{x}_0$

Take home massage

- The idea of Total Variation turns image denoising PB into an optimization PB.
- The data fidelity term and image prior term is widely used as a common optimization structure for other image processing tasks.