Variational Denoising Networks: Toward Blind Noise Modeling and Removal (NeurIPS 2019)

Zongsheng Yue[1]; Hongwei Yong[2]; Qian Zhao[1]; Lei Zhang[2]; Deyu Meng[1]
[1]Xi'an Jiaotong University, [2]The Hong Kong Polytechnic University

Contribution

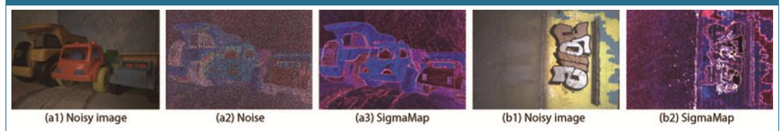

Figure 1: The noise variance map predicted by our proposed VDN on SIDD and DN benchmarks. (a1)-(a3): The noisy image, real noise ($|y-x|$) and noise variance map of one typical image of SIDD validation dataset. (b1-b2): The noisy image and predicted noise variance map of one typical DND image.

- Our method is capable of simultaneously implementing both noise estimation and blind image denoising in a unique Bayesian framework.
- The noise distribution is modeled as a general non-i.i.d. configurations with spatial relevance across the image, which evidently better complies with the heterogeneous real noise beyond the conventional i.i.d. assumption.
- In special case, our method can degenerate as MSE loss generally used in traditional deep learning methods, which provides a new understanding to explain why they incline to overfit noise bias in training data.

Full Bayesian Model

Training set $D=\{\boldsymbol{y}_j,\boldsymbol{x}_j\}_{j=1}^n$, $\boldsymbol{y}_j$: Noisy image, $\boldsymbol{x}_j$: Simulated clean image.

For any $\{\boldsymbol{y},\boldsymbol{x}\}\in D$, we assumed the following generation process:

$$y_i\sim N(y_i|z_i,\sigma_i^2), i=1,2,\cdots,d$$

Conjugate prior for $\boldsymbol{z}$:

$$z_i\sim N(z_i|x_i,\varepsilon_0^2), i=1,2,\cdots,d$$

Conjugate prior for $\boldsymbol{\sigma}$:

$$\sigma_i^2\sim IG\left(\sigma_i^2\middle|\frac{p^2}{2}-1,\frac{p^2\xi_i}{2}\right), i=1,2,\cdots,d$$

Where $\xi=\mathcal{G}((\boldsymbol{y}-\boldsymbol{x})^2;p)$.

Goal (Posterior Inference):

$$p(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y}) \xleftarrow{\text{approximate}} q(z,\sigma^2|y)$$

Likelihood decomposition:

$$\log p(\boldsymbol{y};\boldsymbol{z},\boldsymbol{\sigma}^2)=\underbrace{\mathcal{L}(\boldsymbol{z},\boldsymbol{\sigma}^2;\boldsymbol{y})}_{\text{Variational Lower Bound}}+\underbrace{D_{KL}(q(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y})||p(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y}))}_{\text{Non-negative}}$$

$$\log p(\boldsymbol{y};\boldsymbol{z},\boldsymbol{\sigma}^2)\geq\mathcal{L}(\boldsymbol{z},\boldsymbol{\sigma}^2;\boldsymbol{y})$$

Objective Function:

$$\max\mathcal{L}(\boldsymbol{z},\boldsymbol{\sigma}^2;\boldsymbol{y})$$

Variational Optimization

Conditional independence assumption:

$$q(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y})=q(\boldsymbol{z}|\boldsymbol{y})q(\boldsymbol{\sigma}^2|\boldsymbol{y})$$

Variational posterior form:

$$q(z|y)=\prod\nolimits_i N(z_i|\mu_i,m_i^2), q(\sigma^2|y)=\prod\nolimits_i IG(\sigma_i^2|\alpha_i,\beta_i)$$

Analytical lower bound:

$$\mathcal{L}(z,\sigma^2;y)=E_{q(z,\sigma^2|y)}[\log p(y|z,\sigma^2)]-D_{KL}(q(z|y)||p(z))-D_{KL}(q(\sigma^2|y)||p(\sigma^2))$$

where

$$E_{q(z,\sigma^2|y)}[\log p(y|z,\sigma^2)]=\sum\nolimits_i\left\{-\frac{1}{2}\underbrace{(\log\beta_i-\psi(\alpha_i))}_{E[\log\sigma_i^2]}-\frac{1}{2}\underbrace{\frac{\alpha_i}{\beta_i}}_{E\left[\frac{1}{\sigma_i^2}\right]}[(y_i-\mu_i)^2+m_i^2]\right\}$$

$$D_{KL}(q(z|y)||p(z))=\sum\nolimits_i\left\{\underbrace{\frac{(\mu_i-x_i)^2}{2\varepsilon_0^2}}_{\text{MSE}}+\frac{1}{2}\left[\frac{m_i^2}{\varepsilon_0^2}-\log\frac{m_i^2}{\varepsilon_0^2}-1\right]\right\}$$

$$D_{KL}(q(\sigma^2|y)||p(\sigma^2))=\sum\nolimits_i(\alpha_i-\alpha_i^0)\psi(\alpha_i)+[\log\Gamma(\alpha_i^0)-\log\Gamma(\alpha_i)]$$

$$+\alpha_i^0(\log\beta_i-\log\beta_i^0)+\alpha_i\left(\frac{\beta_i^0}{\beta_i}-1\right)$$

Network Learning

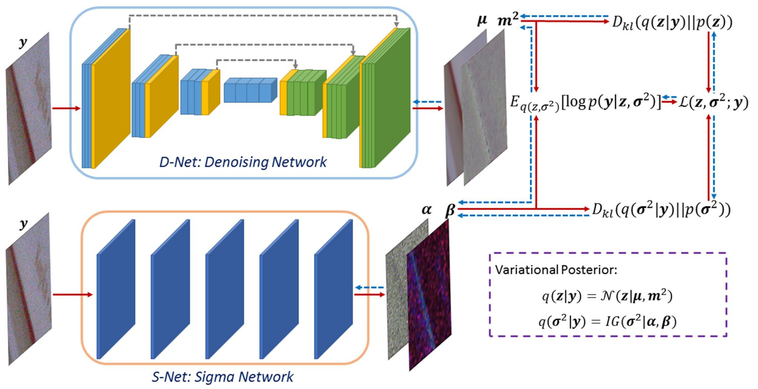

Figure2: The architecture of the proposed deep variational inference network. The red solid lines denote the forward process, and the blue dotted lines mark the gradient flow direction in the BD algorithm.

It should be noted that our proposed method is a general framework, most of the commonly used network architecture in image restoration can also be easily substituted.

Non-IID Gaussian Denoising

Noise generation: $\boldsymbol{n}=\boldsymbol{n}^1\odot\boldsymbol{M}, n_{ij}^1\sim N(n_{ij}^1|0,1)$

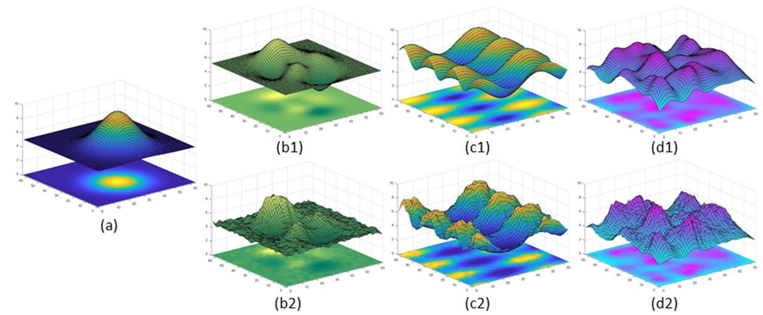

Figure 3: (a) The variance map M for noise generation in training data. (b1)-(d1): Three different Ms on the testing data in Case 1-3. (b2)-(d2): Corresponding predicted Ms by our method on the testing data.

Table 1: The PSNR results of all competing methods on the three groups of test datasets.

Cases	Datasets	Methods								
		CBM3D	WNNM	NCSR	MLP	DnCNN	FFDNet	FFDNet+	UDNet	VDN
Case 1	Set5	27.76	26.53	26.62	27.26	29.87	30.16	30.15	28.13	30.39
	LIVE1	26.58	25.27	24.96	25.71	28.81	28.99	28.96	27.19	29.22
	BSD68	26.51	25.13	24.96	25.58	28.72	28.78	28.77	27.13	29.02
Case 2	Set5	26.34	24.61	25.76	25.73	29.05	29.60	29.56	26.01	29.80
	LIVE1	25.18	23.52	24.08	24.31	28.18	28.58	28.56	25.25	28.82
	BSD68	25.28	23.52	24.27	24.30	28.14	28.43	28.42	25.13	28.67
Case 3	Set5	27.88	26.07	26.84	26.88	29.17	29.54	29.49	27.54	29.74
	LIVE1	26.50	24.67	24.96	25.26	28.15	28.39	28.38	26.48	28.65
	BSD68	26.44	24.60	24.95	25.10	28.10	28.22	28.20	26.44	28.46

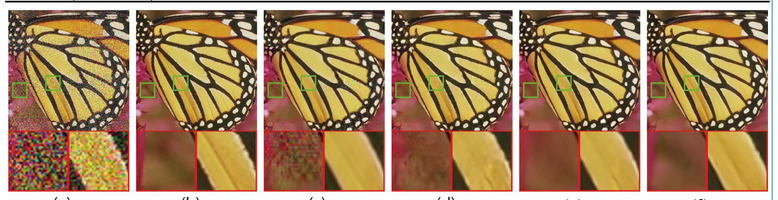

Figure 4:Image Denoising results of a typical test image in Case 2. (a) Noisy image, (b) Groundtruth, (c) CBM3D (24.63 Db), (d) DnCNN (27.83 dB), (e) FFDNet (28.06 dB), (f) VDN (28.32 dB).

Real Benchmark Denoising

Table 2: The comparison results of different methods on SIDD Benchmark and Validation dataset.

Datasets	SIDD Benchmark						SIDD Validation		
Methods	CBM3D	WNNM	MLP	DnCNN	CBDNet	VDN	DnCNN	CBDNet	VDN
PSNR	25.65	25.78	24.71	23.66	33.28	**39.23**	38.41	38.68	**39.28**
SSIM	0.685	0.809	0.641	0.583	0.868	**0.971**	**0.909**	0.901	**0.909**

Table 3: The comparison results of different methods DND Benchmark.

Methods	CBM3D	WNNM	NCSR	MLP	DnCNN	FFDNet	CBDNet	VDN
PSNR	34.51	34.67	34.05	34.23	37.90	37.61	38.06	**39.38**
SSIM	0.8507	0.8646	0.8351	0.8331	0.9430	0.9415	0.9421	**0.9518**

Table 4: Hyper-parameters (ε_0^2 and p) analysis on the SIDD validation dataset.

ε_0^2	1e-4	1e-5	1e-6	1e-7	1e-8	MSE
PSNR	38.89	39.20	**39.28**	39.05	39.03	39.01
SSIM	0.9046	0.9079	**0.9086**	0.9064	0.9063	0.9061

p	5	7	11	15	19
PSNR	39.26	**39.28**	39.26	39.24	39.24
SSIM	**0.9089**	0.9086	0.9086	0.9079	0.9079

Contact

[name] Zongsheng Yue
[organization] Xi'an Jiaotong University
[address] Xi'an, Shaanxi province, China
[email] zsy20110806207@stu.xjtu.edu.cn; zsyzam@gmail.com;

Source Code