

SELF-SUPERVISED BAYESIAN DEEP LEARNING FOR IMAGE DENOISING

(SUPPLEMENTAL MATERIALS)

Anonymous authors

Paper under double-blind review

1 PROOF OF PROPOSITION 1

Recall that the KL divergence between $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$ and $p(\boldsymbol{\theta}|\mathbf{y})$ can be rewritten as

$$\begin{aligned} & \text{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})\|p(\boldsymbol{\theta}|\mathbf{y})) \\ &= \text{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})\|p(\boldsymbol{\theta})) - \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})} \log p(\mathbf{y}|\boldsymbol{\theta}) + \text{const.} \end{aligned} \quad (1)$$

Since $p(\boldsymbol{\theta}) \sim \prod_i \exp(\frac{-\theta_i^2}{2\sigma_i^2})$ and $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma}) \sim \prod_i \exp(\frac{-(\theta_i - \mu_i)^2}{2\sigma_i^2})$, we have

$$\begin{aligned} & \text{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})\|p(\boldsymbol{\theta})) \\ &= \sum_i \text{KL}(q(\theta_i|\mu_i, \sigma_i)\|p(\theta_i)) \\ &= \frac{1}{2\bar{\sigma}^2}(\boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2) - \sum_i \log \sigma_i + \text{const.} \end{aligned} \quad (2)$$

Given the dataset $\{(\hat{\mathbf{y}}, \bar{\mathbf{y}})\}$ generated from \mathbf{y} using the Bernoulli mask \mathbf{b} , $\log p(\mathbf{y}|\boldsymbol{\theta})$ equals to

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{(\hat{\mathbf{y}}, \bar{\mathbf{y}})} p(\bar{\mathbf{y}}|\boldsymbol{\theta}, \hat{\mathbf{y}}). \quad (3)$$

On the other hand,

$$\bar{\mathbf{y}} = (\mathbf{1} - \mathbf{b}) \circ \mathbf{y} = (\mathbf{1} - \mathbf{b}) \circ \mathbf{x} + (\mathbf{1} - \mathbf{b}) \circ \mathbf{n} = (\mathbf{1} - \mathbf{b}) \circ \mathcal{F}_{\boldsymbol{\theta}}(\hat{\mathbf{y}}) + (\mathbf{1} - \mathbf{b}) \circ \mathbf{n}, \quad (4)$$

and $p(\mathbf{n}) \sim \prod_i \exp(\frac{-n_i^2}{2\sigma_i^2})$, thus

$$p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{(\hat{\mathbf{y}}, \bar{\mathbf{y}})} p(\bar{\mathbf{y}}|\boldsymbol{\theta}, \hat{\mathbf{y}}) = -\frac{1}{2\bar{\sigma}} \sum_{\mathbf{b} \sim \mathcal{B}(p)} \|(\mathbf{1} - \mathbf{b}) \circ (\mathcal{F}_{\boldsymbol{\theta}}(\hat{\mathbf{y}}) - \mathbf{y})\|_2^2 + \text{const.} \quad (5)$$

Finally, we obtain

$$\begin{aligned} & \min_{\boldsymbol{\mu}, \boldsymbol{\sigma}} \text{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})\|p(\boldsymbol{\theta}|\mathbf{y})) \\ &= \min_{\boldsymbol{\mu}, \boldsymbol{\sigma}} \sum_{\mathbf{b} \sim \mathcal{B}(p)} \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})} \|(\mathbf{1} - \mathbf{b}) \circ (\mathcal{F}_{\boldsymbol{\theta}}(\hat{\mathbf{y}}) - \mathbf{y})\|_2^2 + \lambda_1(\|\boldsymbol{\mu}\|_2^2 + \|\boldsymbol{\sigma}\|_2^2) - \lambda_2 \sum_i \log \sigma_i, \end{aligned} \quad (6)$$

where $\lambda_1 = \bar{\sigma}^2/\tilde{\sigma}^2$ and $\lambda_2 = 2\bar{\sigma}^2$. The proof is done.

2 MORE EXPERIMENTAL RESULTS

2.1 COMPUTATIONAL COMPLEXITY

Our method is implemented on a NVIDIA TITAN RTX GPU with 24GB Memory. To process the nine images in Set9, whose sizes are listed in Table 1, the total computational time is around 11h35m. In comparison, another unsupervised single-image method, DIP is 7 hours.

2.2 INFLUENCES OF HYPER PARAMETERS

The influences of the values of the hyper parameters $[\lambda_1, \lambda_2]$ in (22) are tested here. We rewrite them as

$$\lambda_1 = \gamma_1 \times \bar{\sigma}^2, \lambda_2 = \gamma_2 \times \bar{\sigma}^2,$$

Table 1: The sizes of images in Set9

Name	hill	baboon, F16, lena, peppers		kodim01, kodim02, kodim03, kodim12	
Size	$3 \times 256 \times 256$	$3 \times 512 \times 512$		$3 \times 768 \times 512$	

γ_1	0.008	0.008	0.01	0.012	0.012
γ_2	0.04	0.05	0.05	0.05	0.06

$\bar{\sigma}=25$	31.62	31.58	31.68	31.54	31.65
$\bar{\sigma}=50$	29.24	29.41	29.39	29.33	29.43

Table 2: The influences of the values of the hyper-parameters on “Set9”.

with $[\gamma_1, \gamma_2] = [0.01, 0.05]$. See Table 2 for the influence of the values of γ_1 and γ_2 on the denoising performance on “Set9”. We can observe that our method is not sensitive to their values.

In Table 3, we show how the Monte Carlo averaging number T impacts the denoising performance on “Set9” for noise level $\bar{\sigma}$ varying from 25 to 100. It can be seen that the performance is improved if more prediction instances are used for averaging, while the gain will saturate after a large value of the Monte Carlo averaging no. T . The value $T = 100$ used by us provides good enough performance.

$\bar{\sigma}$	25	50	75	100
T = 1	29.77	27.48	26.08	24.95
T = 5	31.27	28.93	27.39	26.25
T = 25	31.62	29.32	27.82	26.54
T = 50	31.65	29.35	27.87	26.57
T = 100	31.68	29.39	27.88	26.58
T = 150	31.69	29.40	27.88	26.60

Table 3: PSNR(dB) vs. Monte Carlo averaging no. T on “Set9”

2.3 VISUAL RESULTS

In addition, we also present the visualization of some results from all methods included in the experiments, except SURE (which has no public code available). The results cover the case of Gaussian white noise removal with noise level $\bar{\sigma} = 25, 50$ and the case of real-world image noise removal. See Figure 1, Figure 2 and Figure 3 for the visualization.

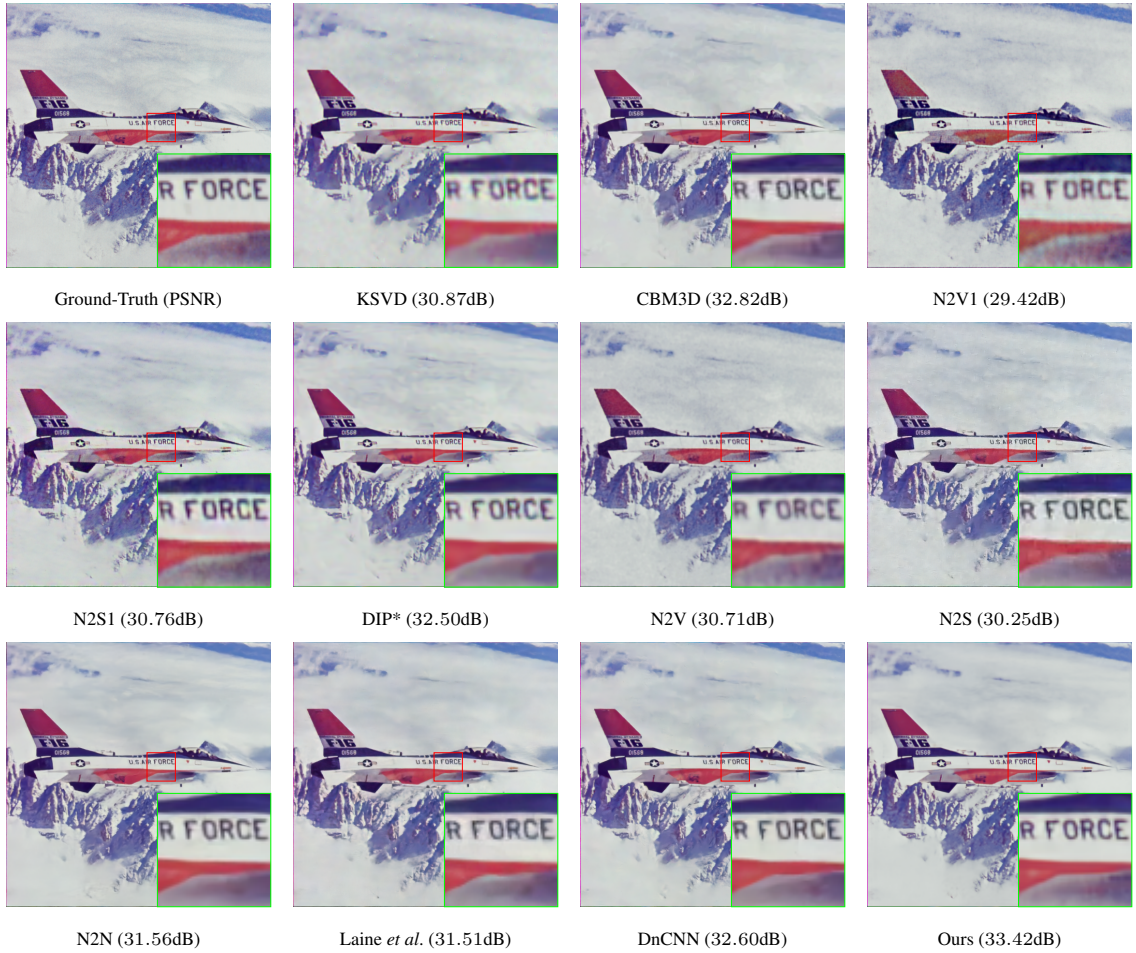


Figure 1: Visual results of removing Gaussian white noise of noise level $\bar{\sigma} = 25$ on image “F16” from Set9.



Figure 2: Visual results of removing Gaussian white noise of noise level $\bar{\sigma} = 50$ on image “test011” from Set68.

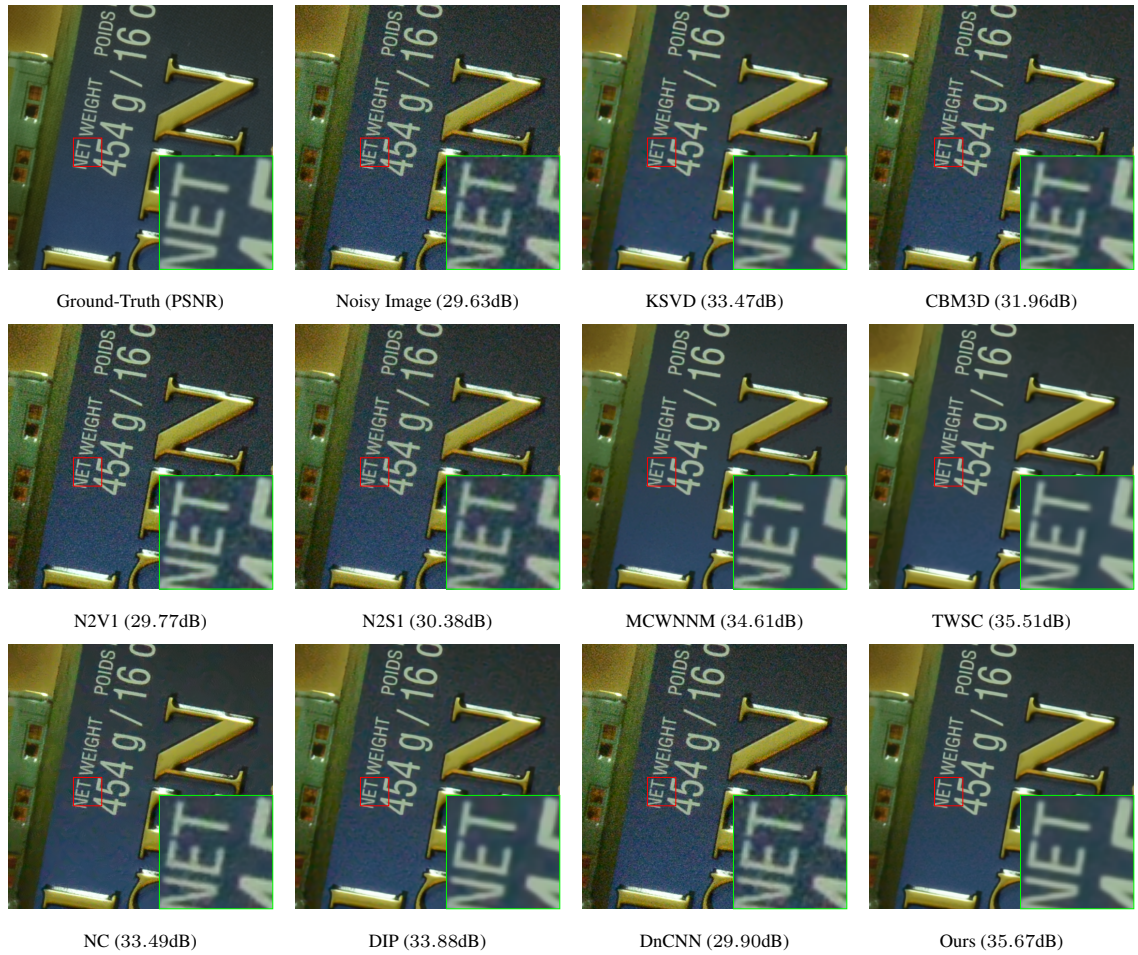


Figure 3: Visual results of denoising the real-word image “d800-iso6400-1” from dataset CC.