NEURAL ADAPTIVE IMAGE DENOISER

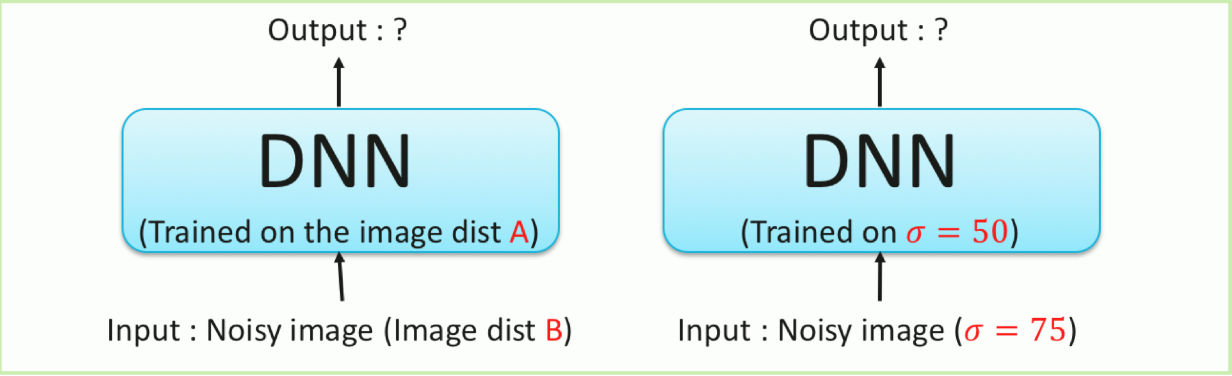
Sungmin Cha and Taesup Moon
Department of Electrical and Computer Engineering
Sungkyunkwan University (SKKU)

Abstract

We propose a novel neural network-based adaptive image denoiser, dubbased as Neural AIDE. Unlike other neural network-based denoisers, which typically apply supervised training to learn a mapping from a noisy patch to a clean patch, we formulate to train a neural network to learn context- based *affine* mappings that get applied to each noisy pixel. Our formulation enables using SURE (Stein's Unbiased Risk Estimator)-like estimated losses of those mappings as empirical risks to minimize. In results, we can combine both supervised training of the network parameters from a separate dataset and *adaptive* fine-tuning of them using the given noisy image subject to denoising. Our algorithm with a plain fully connected architecture is shown to attain a competitive denoising performance on benchmark datasets compared to the strong baselines. Furthermore, Neural AIDE can robustly correct the mismatched noise level in the supervised learning via fine-tuning, of which adaptivity is absent in other neural network-based denoisers.

Introduction

- **Grayscale image denoising**
  - Various denoising methods have been proposed
    - EX) BM3D, WNNM, EPLL, MLP and DnCNN
  - Especially, CNN based image denoising methods recently surpassed the previous state-of-the-arts
- **Drawback of CNN based image denoising methods**
  - Solely based on offline batch training
  - Lacks adaptivity to the given noisy image



- **Problem setting**
  - $x^{n\times n}$: the clean grayscale image, and each pixel $x_i \in [0,255]$
  - Each pixel is corrupted by **an independent additive noise** to result in a noisy pixel $Z_i, i.e., Z_i = x_i + N_i$
  - $\mathbb{E}(N_i) = 0, \mathbb{E}(N_i^2) = \sigma^2$

Estimated loss function for affine denoiser

- **Affine denoiser**

$$\hat{X}_i(Z^{n\times n}) = a(Z^{\setminus i}) \cdot Z_i + b(Z^{\setminus i})$$

- **Estimated loss function for single letter setting**

Suppose $Z = x + N$ with $\mathbb{E}(N) = 0$ and $\mathbb{E}(N^2) = \sigma^2$, and consider a mapping of form $\hat{X}(Z) = aZ + b$. Then,

$$\mathcal{L}(Z,(a,b);\sigma^2) = (Z-(aZ+b))^2 + 2a\sigma^2$$

is an unbiased estimate of $\mathbb{E}\left(x - \hat{X}(Z)\right)^2 + \sigma^2$

- **For each location i, given $Z^{\setminus i}$,**

$$\mathcal{L}\left(Z_i,\left(a(Z^{\setminus i}),b(Z^{\setminus i})\right);\sigma^2\right) = \left(Z_i - \left(a(Z^{\setminus i}) + b(Z^{\setminus i})\right)\right)^2 + 2a\sigma^2$$

is an unbiased estimate of $\mathbb{E}_{Z_i}\left((x_i - \hat{X}_i(Z^{n\times n}))^2 | Z^{\setminus i}\right) + \sigma^2$

Neural AIDE

- **Neural AIDE**

$$\hat{X}_i(Z^{n\times n}) = \mathbf{a}(\mathbf{w}, \mathbf{C}_{k\times k}^{\setminus i}) \cdot Z_i + \mathbf{b}(\mathbf{w}, \mathbf{C}_{k\times k}^{\setminus i})$$

- **Two steps for training Neural AIDE**
  1. Supervised training: minimize

$$\frac{1}{N}\sum_{i=1}^{N}(\tilde{x}_i - (\mathbf{a}(\mathbf{w}, \tilde{\mathbf{C}}_{k\times k}^{\setminus i}) \cdot \tilde{Z}_i + \mathbf{b}(\mathbf{w}, \tilde{\mathbf{C}}_{k\times k}^{\setminus i}))^2$$

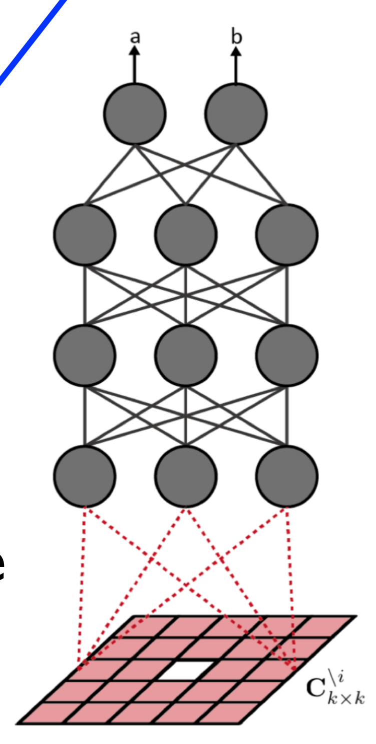
Different from vanilla supervised training

  – $(\tilde{x}, \tilde{Z})$: Collected abundant clean and noisy image pairs

  2. Adaptive training with given noisy image: minimize

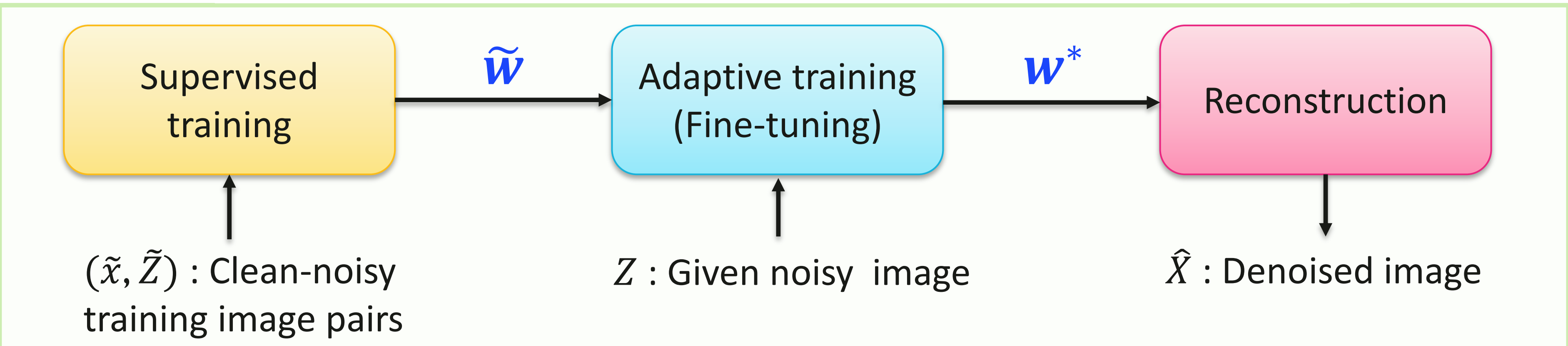
$$\frac{1}{n^2}\sum_{i=1}^{n^2}\mathbf{L}\left(Z_i, (\mathbf{a}(\tilde{\mathbf{w}}, \mathbf{C}_{k\times k}^{\setminus i}), \mathbf{b}(\tilde{\mathbf{w}}, \mathbf{C}_{k\times k}^{\setminus i})); \sigma^2\right)$$

  – Z : Given noisy image



- **The reconstruction at location i by Neural AIDE**

$$\hat{X}_{i,\text{Neural AIDE}}(Z^{n\times n}) = \mathbf{a}(\mathbf{w}^*, \mathbf{C}_{k\times k}^{\setminus i}) \cdot Z_i + \mathbf{b}(\mathbf{w}^*, \mathbf{C}_{k\times k}^{\setminus i})$$



Experimental results

- **Experimental settings**
  - Training data : 2000 training images
  - Test data : the 11 standard benchmark images, the 68 Berkeley test images
  - Context size = 17x17 - 1
  - Model : 9 FC layer with 512 nodes
  - Keras with tensorflow backend

The 11 standard benchmark images

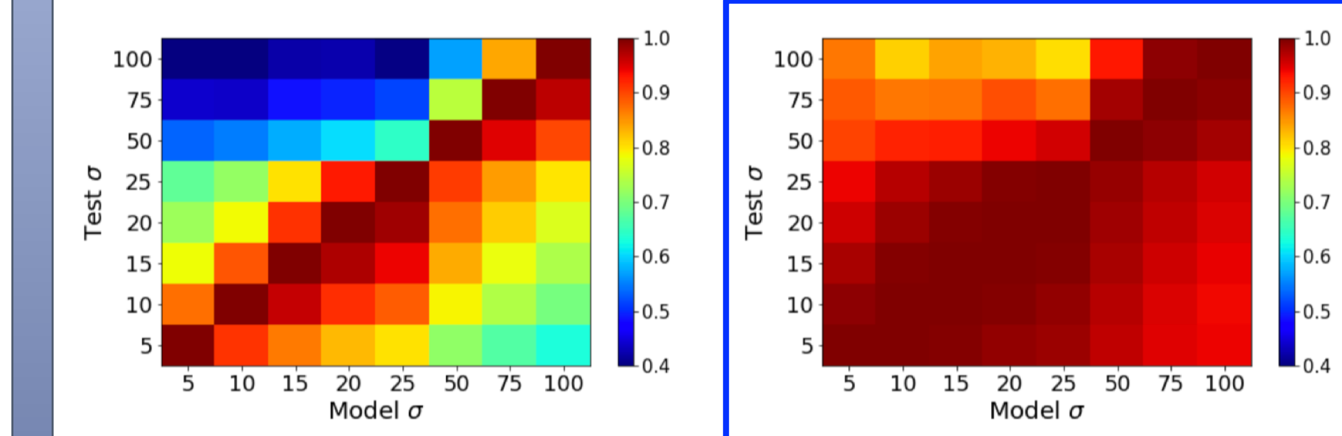
- **Test results**

Trained on $\sigma = [0,50]$

σ	BM3D	MLP	EPLL	WNNM	DnCNN-B	Dn-CNN-S	N-AIDE$_S$	N-AIDE$_{S+FT}$
5	38.23	-	37.90	38.45	37.80	-	38.21	**38.47**
10	34.69	34.45	34.26	**34.95**	34.66	34.88	34.71	34.92
15	32.74	-	32.27	32.99	32.86	**33.02**	32.79	32.98
20	31.40	-	30.90	31.63	31.60	**31.67**	31.43	31.63
25	30.33	30.25	29.79	30.57	30.55	**30.62**	30.32	30.53
50	27.08	-	26.52	27.39	27.43	**27.39**	26.98	27.23
75	25.21	-	24.62	**25.51**	18.62	25.38	25.00	25.27
100	23.96	-	23.42	**24.27**	14.04	-	23.73	23.95

Test noise level is outside the training noise range

- **Test results for mismatched cases**
  - Mismatched models



(a) PSNR of N-AIDE$_S$ (b) PSNR of N-AIDE$_{S+FT}$

Neural AIDE considerably preserve PSNR in mismatched cases!

In mismatched case
DnCNN : 6.76dB↓
NAIDE : 0.5dB↓
NAIDE(Doubly) : 0.66dB↓

  - Doubly mismatched case (Test σ = 75)

Model σ for N-AIDE$_S$	5	10	15	20	25	50
Fine-tuning $\sigma = 75$	22.47	21.99	22.04	22.65	22.12	24.77
Fine-tuning $\sigma = 100$	22.27	21.05	21.64	22.04	21.56	24.61

The 68 Berkeley test images

σ	BM3D	MLP	EPLL	WNNM	DnCNN-B	Dn-CNN-S	N-AIDE$_S$	N-AIDE$_{S+FT}$
5	37.58	-	37.55	37.76	37.65	-	37.81	**37.88**
10	33.31	33.49	33.37	33.55	33.71	**33.86**	33.66	33.75
15	31.07	-	31.19	31.32	31.60	**31.72**	31.47	31.57
20	29.61	-	29.73	29.83	30.19	**30.25**	30.02	30.13
25	28.56	28.95	28.67	28.81	29.15	**29.22**	28.95	29.06
50	25.60	-	25.66	25.89	26.20	**26.21**	25.90	26.03
75	24.19	-	24.09	24.36	18.68	**24.62**	24.31	24.46
100	23.23	-	23.05	23.38	14.29	-	23.24	**23.41**