Image denoising

FDDNet, VDNet, VDID

Chaudhry Abdullah, Pezzulla Simone

abdullah.chaudhry@stud.unifi.it, simone.pezzulla@stud.unifi.it

Elaborazione e Protezione delle immagini

Department of Information Engineering
University of Florence
Via di Santa Marta 3, Florence, Italy

FFDNet [1]

Kai Zhang, Wangmeng Zuo, Senior Member, IEEE, and Lei Zhang, Fellow, IEEE

Introduction

- **FFDNet**: CNN for fast, flexible non-blind and robust denoising.
- **Main techniques** used:
  - Use of noise level map as input
  - Denoising in downsampled sub-images space

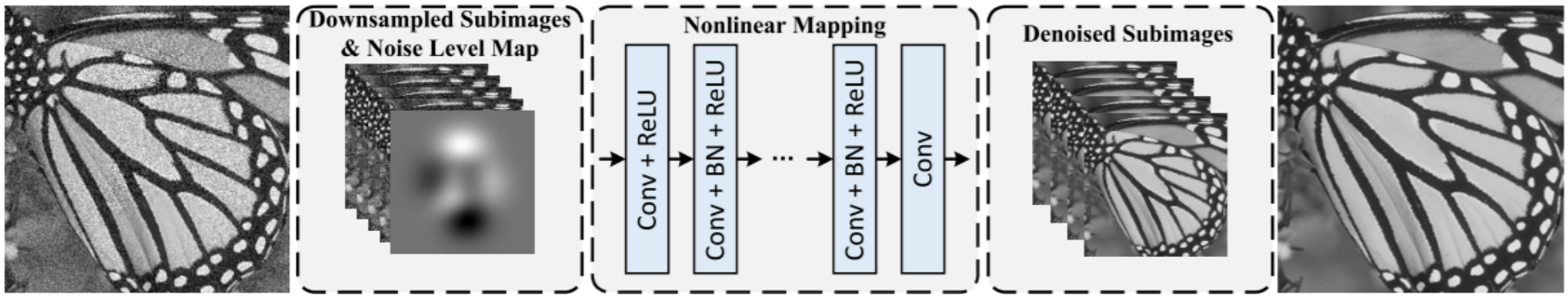


Figure 2: Model Architecture

Model Architecture - Input image

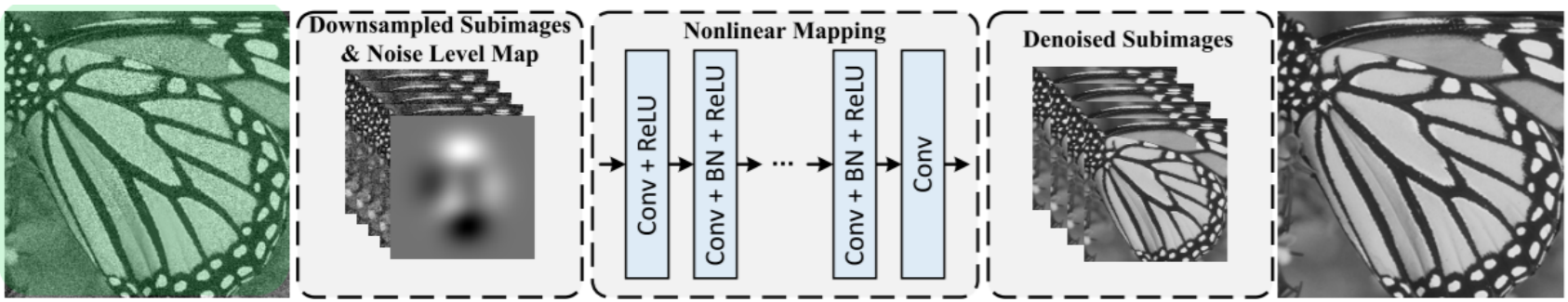


Figure 3: Input Image

Images corrupted by:

1. AWGN
2. Spatially variant AWGN
3. Real-world noise (signal-dependent, non-Gaussian and spatially variant)

Non-Blind Model (FFDNet) vs. Blind Model

The non-blind model has:

- Better generalization ability
- Better performance for AWGN
- Wider application range using variable splitting algorithms:
  - Deblurring
  - SISR
  - Image inpainting

Clipping vs Un-clipping

Clipping: The noisy image is clipped into the range of 0-255 (quantized into 8-bit format).

AWGN

Most denoising methods assume the noise is ideal AWGN, and the clipping of noisy input would make the noise characteristics deviate from being AWGN.

Real noise

1. Real noisy images are always integer-valued and range-limited, clipping setting of noisy image makes the data more realistic.
2. When the noise level is high, the noise will be not zero-mean any more due to clipping effects.
3. This will lead to unreliable denoiser for plugging into the variable splitting algorithms to solve other image restoration problems.

FFDNet in this work refers to the model trained on images without clipping or quantization.

Model Architecture - Noise level map

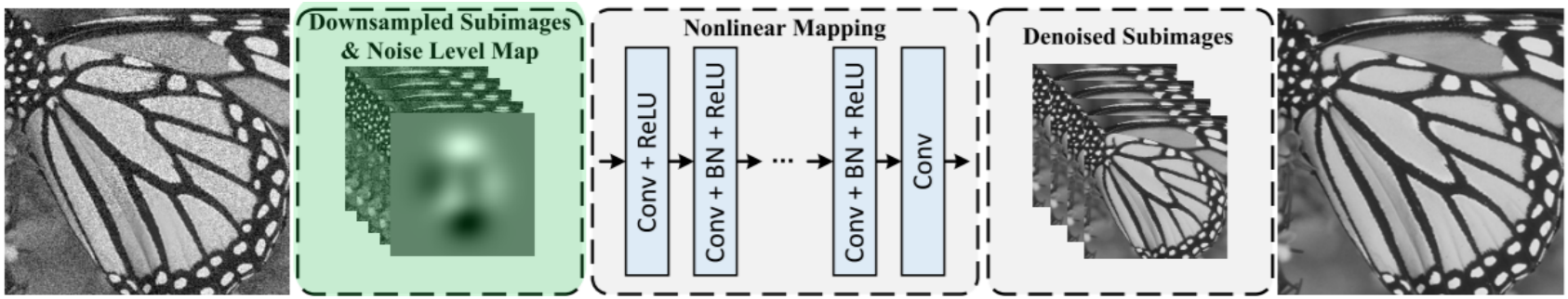


Figure 4: Noise Level Map

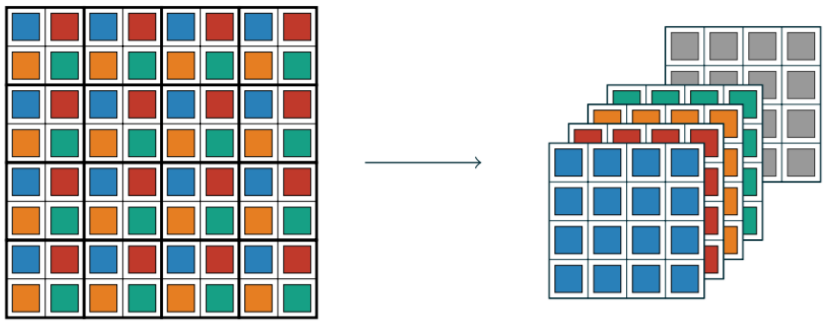
$$\textit{FFDNet} \quad \rightarrow \mathbf{x} = F(\mathbf{y}, \mathbf{M}; \theta)$$

$$\textit{Other models} \quad \rightarrow \mathbf{x} = F(\mathbf{y}; \theta_\sigma)$$

By introducing a noise level map as input, it is natural to expect that the model performs well when the noise level map matches the ground-truth one of noisy input.
Furthermore, the noise level map should also play the role of controlling the **trade-off** between noise reduction and detail preservation.

Model Architecture - Downsampling

The network first reorganizes the pixels of an n_{ch} x h x w input image I into a lower resolution image of size $4n_{ch}$ x $h/2$ x $w/2$.



$\mathbf{I}$

Input

$\mathbf{F^0}$

Subimages and noise map

- Accelerates the training and testing speed without reducing denoising capacity
- Reduce memory burden
- Enlarges the receptive field

Model Architecture - Nonlinear Mapping

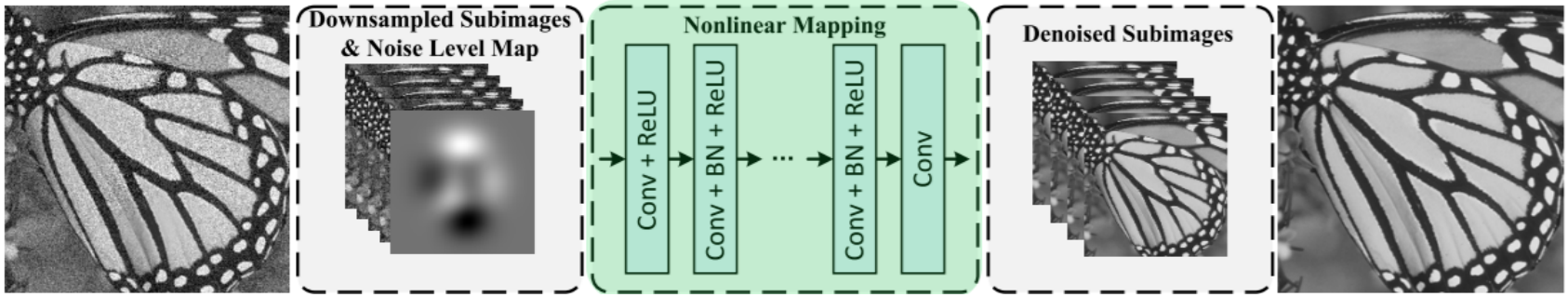


Figure 5: Network Architecture

- Each layer is composed of three types of operations:
  - Convolution
  - Rectified Linear Units
  - Batch Normalization
- Zero-padding is employed to keep the size of feature maps unchanged after each convolution.
- Different settings for grayscale and color image ($D_{RGB} < D_{grey}$ with $W_{RGB} > W_{grey}$).
- $\mathcal{L}(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \|F(y_i, M_i; \theta) - x_i\|^2$

Model Architecture - Nonlinear Mapping (Cont.)

The use of both **M** and **y** as input also increases the difficulty to train the model.
The model may give rise to visual artifacts especially when the input noise level is much higher than the ground-truth one.

One possible solution to avoid this is to regularize the convolution filters through **orthogonal regularization** [2], [3], [4].

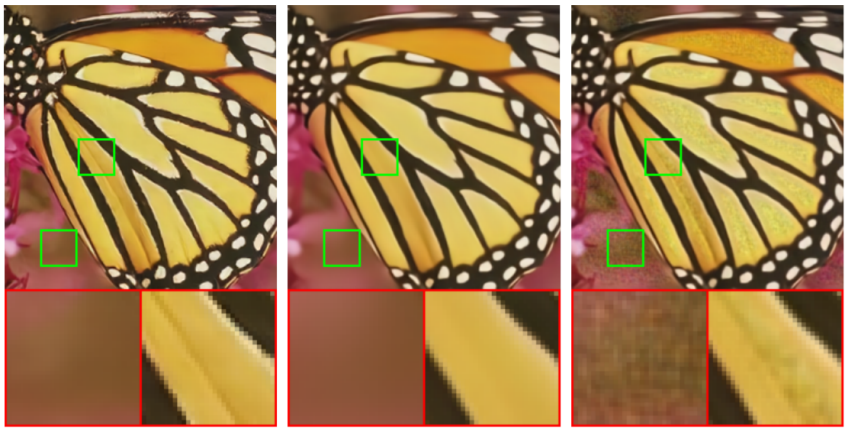


Figure 6: The input is a noisy image with noise level 25. (a) Result with matched noise level 25. (b) and (c) Result with mismatched noise level 60.

Model Architecture - Upscaling

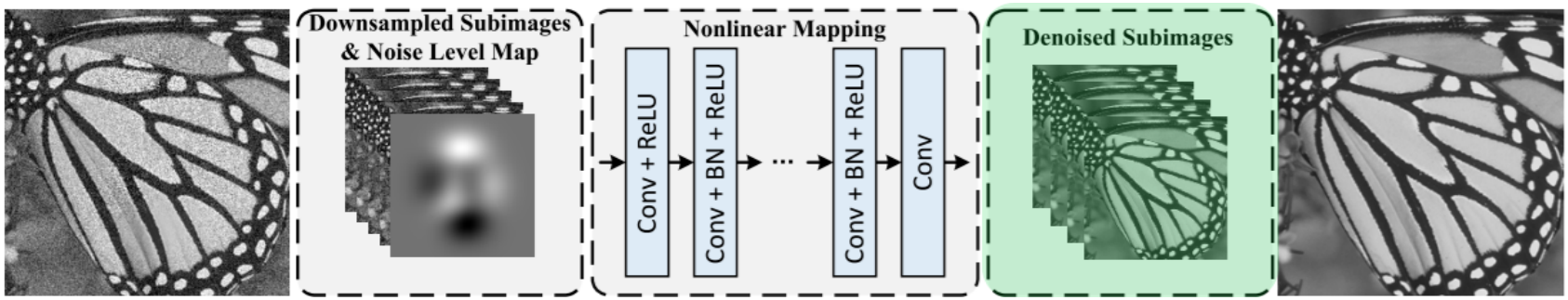


Figure 7: Upscaling

After the last convolution layer, an upscaling operation is applied as the reverse operator of the downsampling operator applied in the input stage to produce the estimated clean image **x** of size $W \times H \times C$

Model Architecture - Residual vs Non-Residual

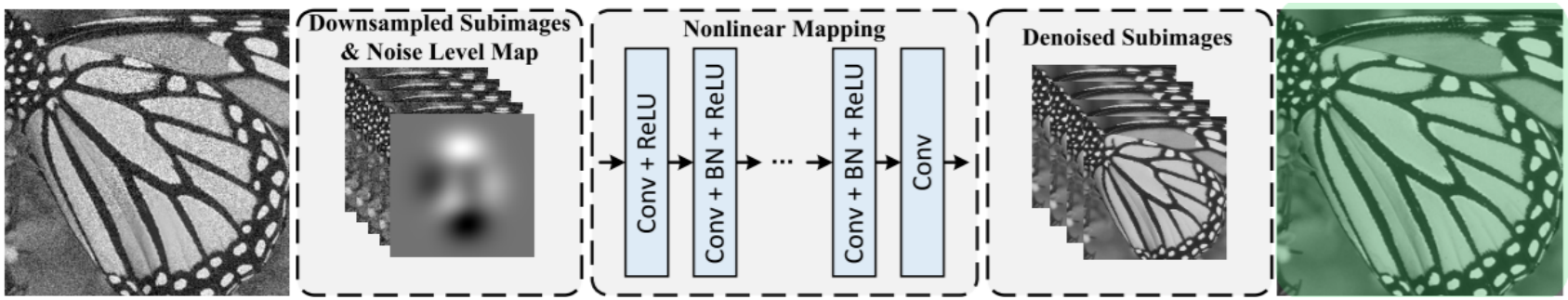


Figure 8: Non-Residual model

1. A Non-residual denoising network gains most when a single noise level is considered (FFDNet considers a wide range of noise level).
2. When a network is moderately deep it is feasible to train a plain network without the residual learning strategy by using advanced CNN training and design techniques such as ReLU[5], BN[6] and Adam[7], reaching nearly the same performance to that with residual learning.

For simplicity FFDNet do not use residual learning.

Dataset generation (Training)

Training: AWGN of noise level $\sigma \in [0, 75]$, the noise level map is **uniform**.

Why AWGN?

- AWGN is a natural choice when there is no specific prior information on noise source.
- A CNN has the **local connectivity property**: output pixel is determined by the local noisy input and local noise level. Hence, the FFDNet naturally has the ability to **handle spatially variant noise** by specifying a non-uniform noise level map.
- **Real-world noise** can be approximated as locally AWGN.

Experiments - spatially invariant AWGN

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

Figure 9: The PSNR(DB) results of different methods on Set12 dataset

Experiments - spatially invariant AWGN (Cont.)

Methods	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$	$\sigma = 60$
DCGRF	31.35	28.67	27.08	25.38	24.45
RBDN	31.05	28.77	27.31	25.80	23.25
FFDNet-Clip	31.68	29.25	27.75	26.25	25.51

Figure 10: (clipping) The average PSNR(DB) results on Clip300

Datasets	Methods	σ=15	σ=25	σ=35	σ=50	σ=75
CBSD68	CBM3D	33.52	30.71	28.89	27.38	25.74
	CDnCNN	33.89	31.23	29.58	27.92	24.47
	FFDNet	33.87	31.21	29.58	27.96	26.24
Kodak24	CBM3D	34.28	31.68	29.90	28.46	26.82
	CDnCNN	34.48	32.03	30.46	28.85	25.04
	FFDNet	34.63	32.13	30.57	28.98	27.27
McMaster	CBM3D	34.06	31.66	29.92	28.51	26.79
	CDnCNN	33.44	31.51	30.14	28.61	25.10
	FFDNet	34.66	32.35	30.81	29.18	27.33

Figure 11: (Color) The average PSNR(DB) results of CBM3D, CDnCNN and FFDNet on CBSD68, Kodak24 and McMaster datasets

Experiments - spatially variant AWGN

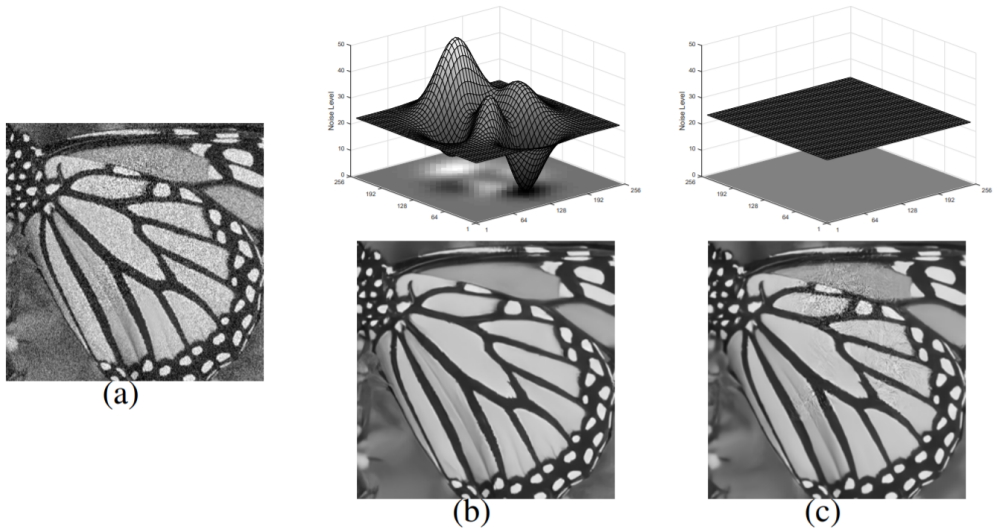


Figure 12: Examples of FFDNet on removing spatially variant AWGN. (a) Noisy image (20.55dB) with spatially variant AWGN. (b) Ground-truth noise level map and corresponding denoised image (30.08dB) by FFDNet; (c) Uniform noise level map constructed by using the mean value of ground-truth noise level map and corresponding denoised image (27.45dB) by FFDNet.

Experiments - Real noise

Interactive strategy to handle real noisy images:

1. Assumption of **spatially invariant** noise usually works well for most real noisy images
   - Employ a set of typical input noise levels to produce multiple outputs, and select the one which has best trade-off
2. **Spatially variant** noise in most real-world images is signal-dependent:
   - Sample several typical regions
   - Apply different noise levels with an interval of 5 and choose the best noise level
   - The noise levels at other regions are then interpolated of distinct colors

FFDNet **compared** with BM3D, DnCNN, DnCNN-B and Noise Clinic

Conclusions - FFDNet

FFDNet for fast, effective and flexible discriminative denoising through **several techniques** such as noise level map as input and denoising in downsampled sub-images space.

- Results on **AWGN**:
  - State-of-the-art results when input noise level matches ground-truth noise level.
  - Robustly control the trade-off between noise reduction and detail preservation.
- Results on **spatially variant AWGN**:
  - Validated the flexibility of FFDNet for handing inhomogeneous noise
- Results on **real noisy images**:
  - FFDNet can deliver perceptually appealing denoising results

Variational Denoising Network[8]

Zongsheng Yue, Hongwei Yong, Qian Zhao, Lei Zhang, Deyu Meng

Introduction

Concept

Use of *variational inference* for non-iid **real noise estimation** and **image denoising** in a unique Bayesian Network.

Introduction

Concept

Use of *variational inference* for non-iid **real noise estimation** and **image denoising** in a unique Bayesian Network.

- **Input**: noisy image y
- **Latent variables**: clean image z and noise variance σ^2
- **Goal**: infer (an approximation of) the posterior

$$p(z, \sigma^2 | y)$$

Use of Networks

- **For clean image** z:
  We consider z with a Gaussian distribution.
  μ and m^2 are parameters to approximate the posterior of z
  - Predicted by **D-Net**

Use of Networks

- **For clean image** z:
  We consider z with a Gaussian distribution.
  μ and m^2 are parameters to approximate the posterior of z
  - Predicted by **D-Net**
- Similarly, **for the noise level** σ^2:
  We consider σ^2 with Inverse Gamma distribution
  α and β are parameters to approximate the posterior of σ^2
  - Predicted by **S-Net**

Network learning

Marginal data likelihood

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

where $\mathcal{L}(z_j, \sigma_j^2; y_j)$ can be expressed in function of μ, m^2, α, β

Network learning

Marginal data likelihood

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

where $\mathcal{L}(z_j, \sigma_j^2; y_j)$ can be expressed in function of μ, m^2, α, β

Objective function

We need to maximize the Lower Bound.

$$\min_{W_D, W_S} -\sum_{j=1}^{n} \mathcal{L}(z_j, \sigma_j^2; y_j)$$

- **Training**: with backpropagation updating both the parameters of D-Net and S-Net simultaneously
- **Test**: we can obtain the clean image and noise distribution from parameters μ, m^2, α, β

Network Architecture

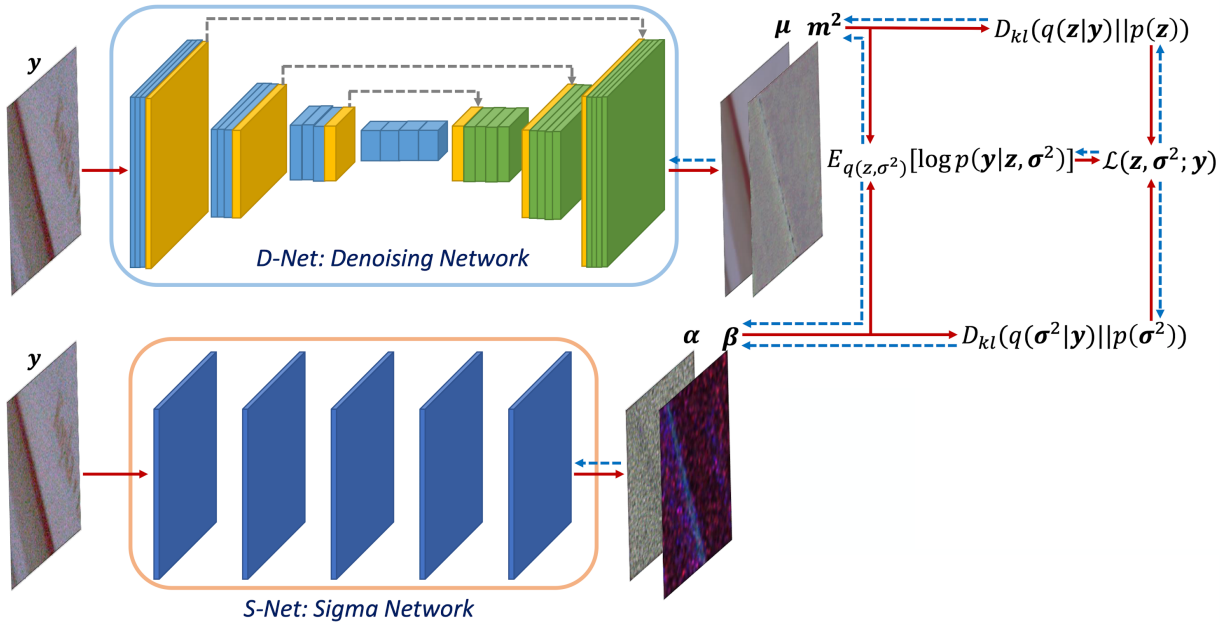


Figure 13: The red solid lines denote the forward process, and the blue dotted lines mark the gradient flow direction in the BP algorithm.

- D-Net: U-Net [9] with depth 4, which contains 4 encoder blocks, 3 decoder blocks, and symmetric skip connection under each scale in order to capture multi-scale information of the image
- S-Net: DnCNN [10] architecture with five layers, and the feature channels of each layer is set as 64.

Experimental Results - Non-i.i.d Gaussian Noise

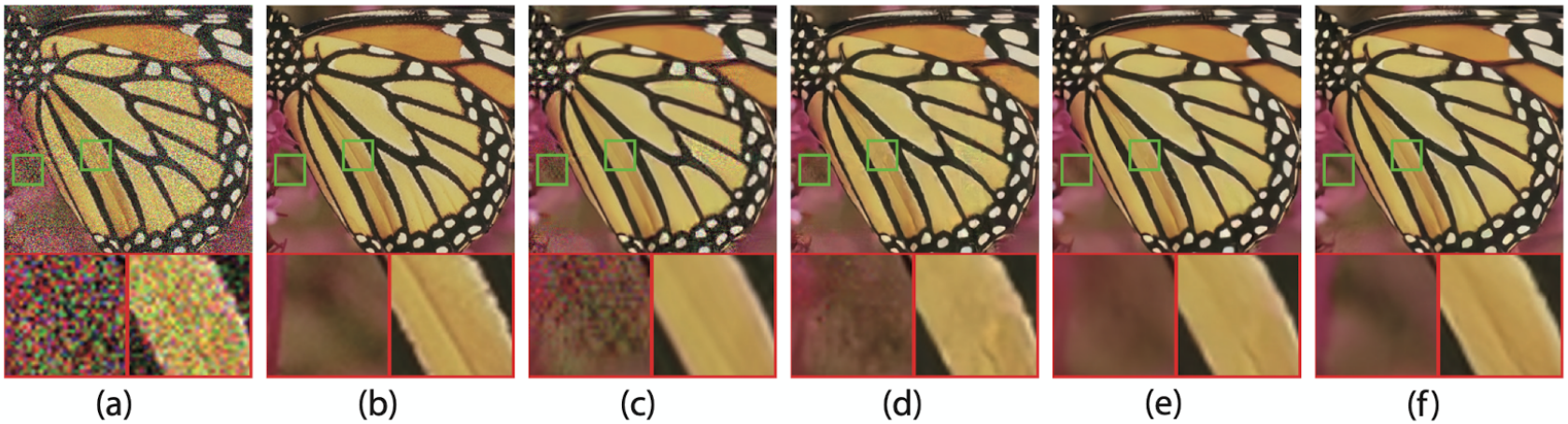


Figure 14: Image denoising results of a typical test image. (a) Noisy image, (b) Groundtruth, (c) CBM3D (24.63dB), (d) DnCNN-B (27.83dB), (e) FFDNet (28.06dB), (f) VDN (28.32dB).

Experimental Results - Non-i.i.d Gaussian Noise

Cases	Datasets	Methods									
		CBM3D	WNNM	NCSR	MLP	DnCNN-B	MemNet	FFDNet	FFDNet$_v$	UDNet	VDN
Case 1	Set5	27.76	26.53	26.62	27.26	29.85	30.10	*30.16*	30.15	28.13	**30.39**
	LIVE1	26.58	25.27	24.96	25.71	28.81	28.96	*28.99*	28.96	27.19	**29.22**
	BSD68	26.51	25.13	24.96	25.58	28.73	28.74	*28.78*	28.77	27.13	**29.02**
Case 2	Set5	26.34	24.61	25.76	25.73	29.04	29.55	*29.60*	29.56	26.01	**29.80**
	LIVE1	25.18	23.52	24.08	24.31	28.18	28.56	*28.58*	28.56	25.25	**28.82**
	BSD68	25.28	23.52	24.27	24.30	28.15	28.36	*28.43*	28.42	25.13	**28.67**
Case 3	Set5	27.88	26.07	26.84	26.88	29.13	29.51	*29.54*	29.49	27.54	**29.74**
	LIVE1	26.50	24.67	24.96	25.26	28.17	28.37	*28.39*	28.38	26.48	**28.65**
	BSD68	26.44	24.60	24.95	25.10	28.11	28.20	*28.22*	28.20	26.44	**28.46**

Figure 15: The PSNR(dB) results of all competing methods on the three groups of test datasets

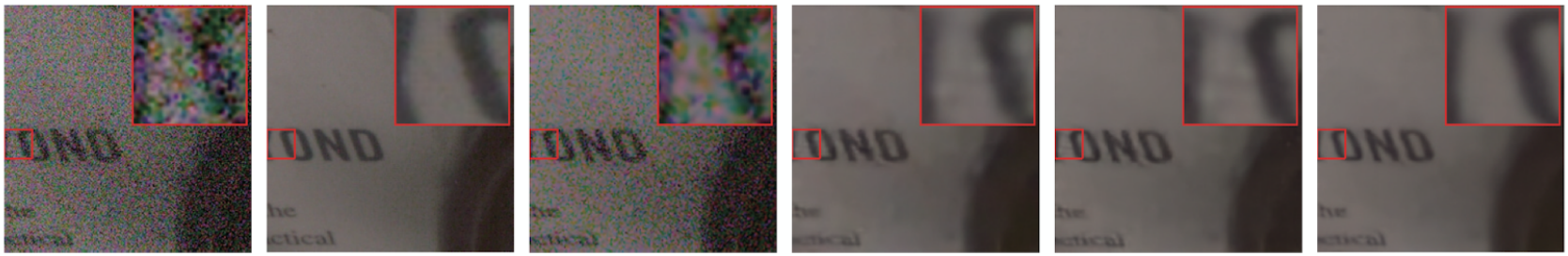
- The VDN **outperforms** other competing methods in all cases
- VDN surpasses FFDNet about 0.25dB averagely even though FFDNet depends on the true noise level information
- WNNM and NCSR are the worst, possibly due to the latter's improper i.i.d. Gaussian noise assumption

Experimental Results - AWGN

Sigma	Datasets	Methods									
		CBM3D	WNNM	NCSR	MLP	DnCNN-B	MemNet	FFDNet	FFDNet$_e$	UDNet	VDN
$\sigma = 15$	Set5	33.42	32.92	32.57	-	34.04	34.18	34.30	*34.31*	34.19	**34.34**
	LIVE1	32.85	31.70	31.46	-	33.72	33.84	**33.96**	**33.96**	33.74	33.94
	BSD68	32.67	31.27	30.84	-	*33.87*	33.76	33.85	33.68	33.76	**33.90**
$\sigma = 25$	Set5	30.92	30.61	30.33	30.55	31.88	31.98	*32.10*	32.09	31.82	**32.24**
	LIVE1	30.05	29.15	29.05	29.16	31.23	31.26	*31.37*	*31.37*	31.09	**31.50**
	BSD68	29.83	28.62	28.35	28.93	*31.22*	31.17	31.21	31.20	31.02	**31.35**
$\sigma = 50$	Set5	28.16	27.58	27.20	27.59	28.95	29.10	*29.25*	*29.25*	28.87	**29.47**
	LIVE1	26.98	26.07	26.06	26.12	27.95	27.99	*28.10*	*28.10*	27.82	**28.36**
	BSD68	26.81	25.86	25.75	26.01	27.91	27.91	*27.95*	*27.95*	27.76	**28.19**

Figure 16: The PSNR(dB) results of all competing methods on AWGN noise cases of three test datasets.

Experimental Results - Real-World Noise



(a) (b) (c) (d) (e) (f)

Figure 17: Denoising results on one typical image in the validation set of SIDD. (a) Noisy image, (b) Simulated "clean" image, (c) WNNM(21.80dB), (d) DnCNN (34.48dB), (e) CBDNet (34.84dB), (d) VDN (35.50dB).

Datasets	SIDD Benchmark						SIDD Validation		
Methods	CBM3D	WNNM	MLP	DnCNN-B	CBDNet	VDN	DnCNN-B	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**

Figure 18: The comparison results of different methods on SIDD benchmark and validation dataset.

Experimental Results - Real-World Noise

...VDN is still the best for DND benchmark dataset

Methods	CBM3D	WNNM	NCSR	MLP	DnCNN-B	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**

Figure 19: The comparison results of all competing methods on DND benchmark dataset.

Conclusion

- New **variational inference algorithm**, namely varitional denoising network (VDN), for blind image denoising has been proposed
- It's a **generative method**, which can easily estimate the noise distribution from the input data
- It can facilitate the study of other low-level vision tasks, such as super-resolution and deblurring.

Variational Deep Image Denoising[11]

Jae Woong Soh, Nam Ik Cho

Introduction

Variational Deep Image Denoiser (VDID)

Bayesian framework that can handle blind scenarios based on the variational approximation of objective functions separating the complicated problem into simpler ones.

Introduction

Variational Deep Image Denoiser (VDID)

Bayesian framework that can handle blind scenarios based on the variational approximation of objective functions separating the complicated problem into simpler ones.

Characteristics

- It handles **both AWGN and real-world noise**
- Trained in an end-to-end scheme without any additional noise information
- Requires fewer parameters than state-of-the-art denoisers

Variational Deep Image Denoising

The objective is formulated in terms of **maximum a posterior** (MAP) inference.
An approximated form of the objective is calculated by introducing a latent variable based on variational Bayes which incorporates the underlying noisy image distribution.

Based on the latent space, VDID can focus on simpler subdistributions of the original problem.

Network architecture

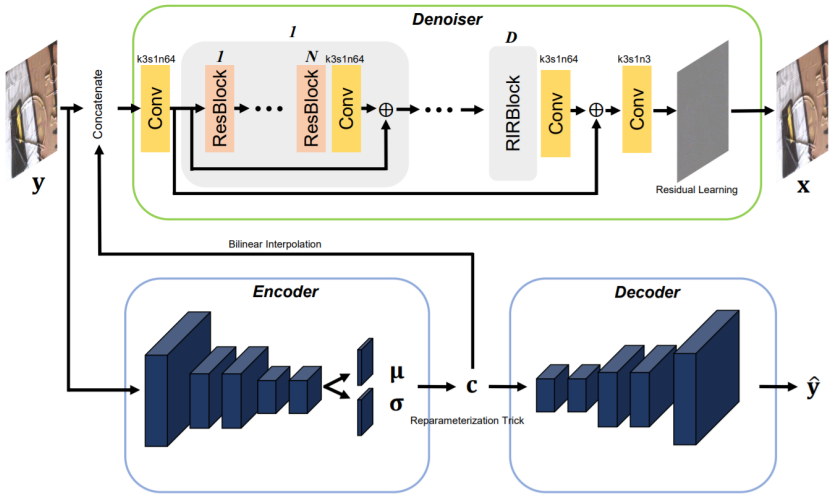


Figure 20: The overall architecture of proposed VDID

- Autoencoder: feedforward convolutional network with a reparametrization trick
- Denoiser is fully convolutional
- ResBlock is adopted as the basic building block
- Skip connections (denoiser only)
- The last convolution layer infers the residual image (noise) instead of the clean image

Loss Terms

- **Mean absolute error** (MAE) between the ground-truth clean image and the inferred output to minimize the distortion

$$\mathcal{L}_{denoise} = ||x - \hat{x}||_1$$

- The **KL divergence** between the posterior and the prior. The latent variable c contains an abstract of the noisy image distribution

$$D_{KL}(q(c|y)||p(c))$$

- The third term is composed by
  - **MAE** between the **noisy input image** and the **decoder's output**
  - **Adversarial loss** [12]: to better learn the noisy image distribution
  - **Noise estimation loss**: σ_N is the noise level and EST($\cdot$) is a simple two-layer CNN.

$$\mathcal{L}_{recon} = ||y - \hat{y}||_1 + \lambda_1 \mathcal{L}_{adv} + \lambda_2 ||\sigma_N - EST(c)||_1$$

Overall Loss

The overall loss is the sum of three terms aforementioned. Where

- The first term is just a naive approach totally relying on the discriminative power of CNN.
- In the second term the KL divergence forces c to have discriminative power over observed noisy images y.
- The third term forces c to include the information on a noisy image.

Experimental Results - AWGN removal

Noise level	Dataset	CBM3D [9]	RED [29]	CDnCNN [48]	FFDNet [49]	UNLNet [24]	VDN [44]	VDID (Ours)
$\sigma_N = 10$	CBSD68	35.91	33.89	36.13	36.14	36.20	36.29	36.34
	Kodak24	36.43	34.73	36.46	36.69	-	36.85	37.02
	Urban100	36.00	34.42	34.61	35.78	-	35.97	36.30
$\sigma_N = 30$	CBSD68	29.73	28.45	30.34	30.32	30.21	30.64	30.64
	Kodak24	30.75	29.53	31.17	31.27	31.18	31.67	31.74
	Urban100	30.36	28.84	30.00	30.53	30.41	31.14	31.41
$\sigma_N = 50$	CBSD68	27.38	26.34	27.95	27.97	27.85	28.33	28.33
	Kodak24	28.46	27.42	28.83	28.98	28.86	29.44	29.49
	Urban100	27.94	26.25	27.59	28.05	27.95	28.86	29.10
$\sigma_N = 70$	CBSD68	26.00	25.09	25.66	26.55	-	26.93	26.94
	Kodak24	27.09	26.16	26.36	27.56	-	28.05	28.10
	Urban100	26.31	24.58	25.24	26.40	-	27.31	27.55

Figure 21: The average PSNR on AWGN denoising.

Experimental Results - Real-Noise removal

Method	Blind/Non-blind	Parameters	PSNR	SSIM
BM3D [9]	Non-blind	-	25.65	0.685
WNNM [15]	Non-blind	-	25.78	0.809
DnCNN [48]	Non-blind	668 K	23.66	0.583
TNRD [8]	Non-blind	27 K	24.73	0.643
CBDNet [16]	Blind	4.4 M	33.28	0.868
RIDNet [4]	Blind	1.5 M	38.71	0.914
VDN [44]	Blind	7.8 M	39.26	0.955
AINDNet+TF [22]	Blind	13.7 M	38.95	0.952
VDID (Ours)	Blind	2.2 M	39.25	0.955
VDID+ (Ours)	Blind	2.2 M	**39.33**	**0.956**

Figure 22: Results on SIDD benchmark on Real-Noise removal.

Method	Blind/Non-blind	Parameters	PSNR	SSIM
BM3D [9]	Non-blind	-	34.51	0.8507
WNNM [15]	Non-blind	-	34.67	0.8646
DnCNN+ [48]	Non-blind	668 K	37.90	0.9430
FFDNet+ [49]	Non-blind	825 K	37.61	0.9415
GCBD [7]	Blind	561 K	35.58	0.9217
CBDNet [16]	Blind	4.4 M	38.06	0.9421
RIDNet [4]	Blind	1.5 M	39.26	0.9528
VDN [44]	Blind	7.8 M	39.38	0.9518
AINDNet(S) [22]	Blind	13.7 M	39.53	**0.9561**
VDID (Ours)	Blind	2.2 M	39.63	0.9528
VDID+ (Ours)	Blind	2.2 M	**39.69**	0.9532

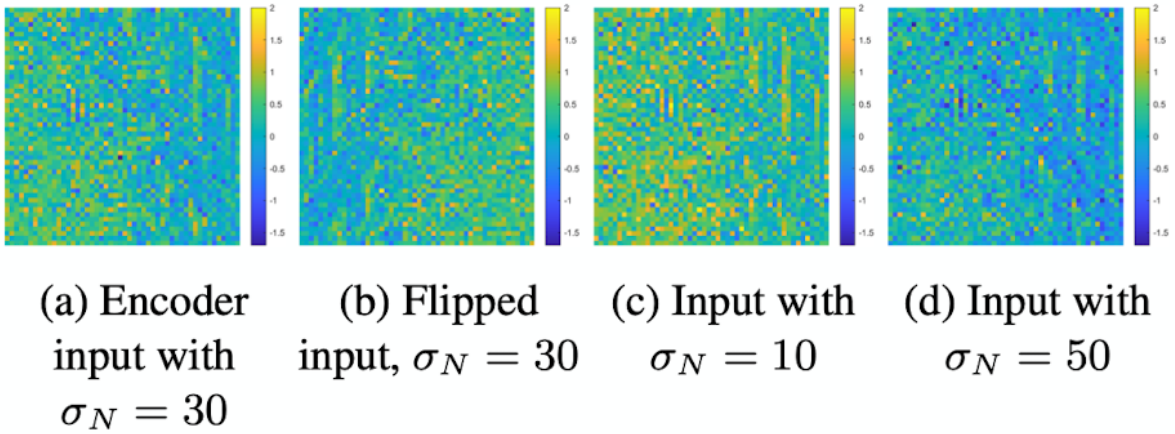
Figure 23: Results on DND benchmark.

Latent Variable Manipulation

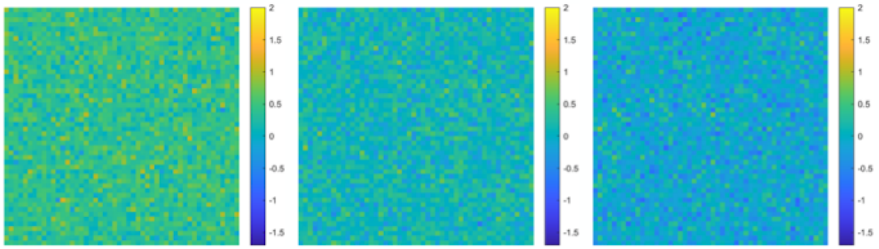
(a) Noisy image $\sigma_N = 30$ 19.72 dB | (b) Restoration of (a) 27.20 dB | (c) Restoration with $\mathbf{c}$ from $\sigma_N = 10$ 23.95 dB | (d) Restoration with $\mathbf{c}$ from $\sigma_N = 50$ 24.93 dB | (e) Restoration with $\mathbf{c}$ from flipped input 26.74 dB

Figure 24: Denoising results depending on c. Noisy image (a) in input to the denoising network while changing the input to the encoder that extracts c. (b)Result of VDID with the same input to the denoiser and encoder. (c)Result when the encoder is given the input with $\sigma_N = 10$ so that c is the latent variable correponding to $\sigma_N = 10$. (d)c correponds to $\sigma_N = 50$. (e)The encoder is given the flipped patch with $\sigma_N = 30$.

Latent Variable Visualization



(a) Encoder input with $\sigma_N = 30$ (b) Flipped input, $\sigma_N = 30$ (c) Input with $\sigma_N = 10$ (d) Input with $\sigma_N = 50$



(e) Flat input, $\sigma_N = 10$ (f) Flat input, $\sigma_N = 30$ (g) Flat input, $\sigma_N = 50$

Figure 25: Visualization of c for the image in Figure 24 and for a flat image with varying σ_N. (a) c from the encoder, when its input is Fig. 24a, (b) when the input is the flip of Fig. 24b, (c) when the input is the same but $\sigma_N = 10$, and (d) $\sigma_N = 50$. (e) c from the encoder given a flat patch with $\sigma_N = 10$, (f) $\sigma_N = 30$, and (g) $\sigma_N = 50$.

Conclusion

- Novel variational approach for image denoising
- With the variational lower bound, the original problem can be divided into separate sub-problems, which eventually relaxes the given problem
- Three parameterized CNNs for the inference problem, achieving state-of-the-art performances in removing both Gaussian and Real-World noises while requiring fewer parameters.

Questions?

Thanks! Any questions?

Bibliography

[1] Kai Zhang, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising". In: *IEEE Transactions on Image Processing* (2018).

[2] keras. *Keras orthogonal initializer*. URL: https://keras.io/api/layers/initializers/.

[3] K. Jia. "Improving training of deep neural networks via singular value bounding". In: *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[4] D. Mishkin and J. Matas. "All you need is a good init". In: *ArXiv e-prints* (2015).

[5] I. Sutskever A. Krizhevsky and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* (2012).

[6] S. Ioffe and C. Szegedy. ""Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning* (2015).

Bibliography (cont.)

[7] D. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *International Conference for Learning Representations* (2015).

[8] Qian Zhao Zongsheng Yue Hongwei Yong. "Variational Denoising Network: Toward Blind Noise Modeling and Removal". In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada* (2019).

[9] Philipp Fischer Olaf Ronneberger and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention* (2015).

[10] Y. Chen K. Zhang W. Zuo. "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising". In: *IEEE Transactions on Image Processing* (2017).

[11] Nam Ik Cho Jae Woong Soh. "Variational Deep Image Denoising". In: *ArXiv e-prints* (2021).

[12] Jean Pouget-Abadie Ian Goodfellow. "Generative adversarial nets". In: *Advances in neural information processing systems* (2014).