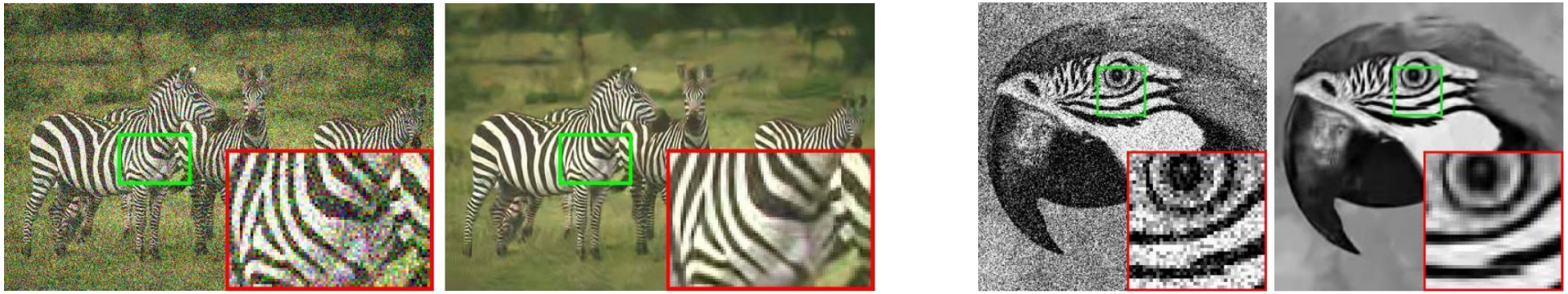
DEEP DENOISING

Geonwoon Jang
Prof. Kyoung Mu Lee

Computer Vision Lab
Dept. of ECE, Seoul National University, Korea
Homepage: https://cv.snu.ac.kr

Intro to Denoising

- Reduction of unwanted signal



- There are many sources of noise in camera system.
  - Electronic devices(CCD, diode, amp., etc.)
  - Thermal motion of charge carrier
  - Demosaicing, digital conversion, transmission

Types of the problem

- The problem of denoising(noise reduction) can be classified in detailed ways

- In target data types:
  - Single image (gray/color)

  - Fast burst images

  - Video

  - Others(medical, fingerprint, etc.)

Types of the problem

- The problem of denoising(noise reduction) can be classified in detailed ways

- In noise types:
  - Synthetic noise
    - Use input noisy image as '**GT clear image**(unseen during inference) **+ noise map**'
    - Because it's very hard to get perfectly aligned (noisy, clear) image pair, it **have been a basic problem** to remove this synthetic noise.
      - **Non-blind** denoising: one knows already knows the noise function, for example the std of additive Gaussian noise is given during training.
      - **Blind** denoising: one doesn't know about the noise function.
  - Real noise
    - However, synthetic noise cannot mimic various real noise sources.
    - For **static scenes**, getting large sequence of data with **fixed camera**, one can generate a clean image. (though this process is very exhaustive)
    - It is **more challenging** because there's **not enough training data pair**.

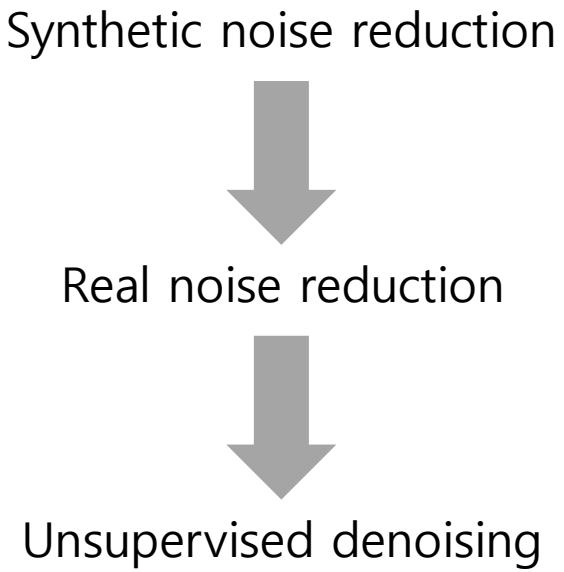
Types of the problem

- The problem of denoising(noise reduction) can be classified in detailed ways

- In learning scheme:
  - Supervised learning
    - For both synthetic and real noise, use same GT input image and noisy image pairs for training.
  - Unsupervised learning
    - For it's generally very **hard to get a pair** noisy and clean image or multiple aligned images,
    - Train **without GT** image
    - There are some minor problem settings for this:
      - Assumes one can get **perfectly-aligned multiple noisy images** from static scene, only their noise map is different.
      - For this, use **multiple** 'GT clear image(unseen) + noise map' for training.

Types of the problem

- Deep-learning based denoising has been becoming practical.



Classical Methods - Smoothing

- One of the most basic traditional denoising method would be smoothing
  - It is basically convolving an image with certain kernel.
  - The kernel can be, for example, the Gaussian:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{|x|^2}{2\sigma^2}}$$

  - or median filter, circle mean, etc.

  - However, the kernel should be chosen according to the noise,
  - and it produces over-smoothed results (also its extended algorithms still does)

original | 3*3 median filtered | 7*7 median filtered | 11*11 median filtered

Classical Methods - Bilateral Filtering

- Bilateral Filtering
  - A kind of **Neighborhood Filtering** method, which means it takes average of neighboring pixel values for a target pixel, with proper weighting.
  - For example, taking average from fixed size of spatial neighbors:

$$v'(x) = \frac{1}{C}\int_{B(x)} v(y) e^{-\frac{|v(y)-v(x)|^2}{h^2}} dy$$

, where B means neighbors of x and $C = \int_{B(x)} e^{-\frac{|v(y)-v(x)|^2}{h^2}} dy$

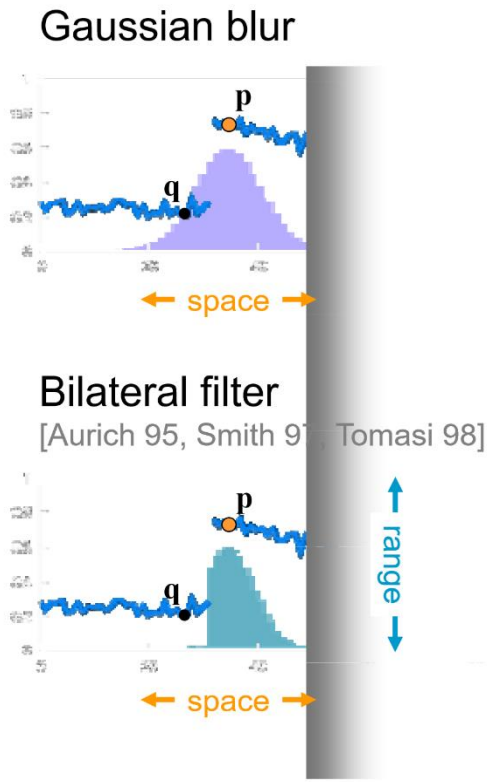
  - **Bilateral Filter** considers pixels globally and weigh them with their distance from target x.

$$v'(x) = \frac{1}{C}\int v(y) e^{-\frac{|y-x|^2}{\rho^2}} e^{-\frac{|v(y)-v(x)|^2}{h^2}} dy$$

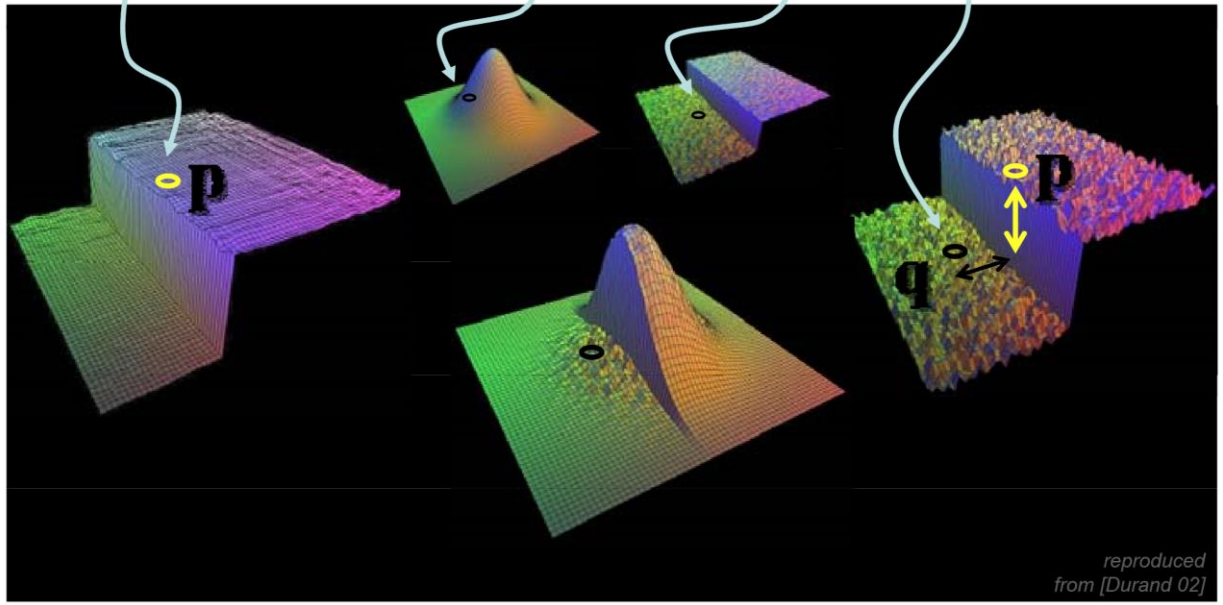
, where $C = \int e^{-\frac{|y-x|^2}{\rho^2}} e^{-\frac{|v(y)-v(x)|^2}{h^2}} dy$

Classical Methods - Bilateral Filtering

- Comparison between smoothing and Bilateral Filtering



$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$



visualizations by Yung-Yu Chuang, Fredo Durand, Ramesh Raskar, Sylvain Paris, Soonmin Bae

- There are some extensions to accelerate this algorithm.

Classical Methods - NL-means

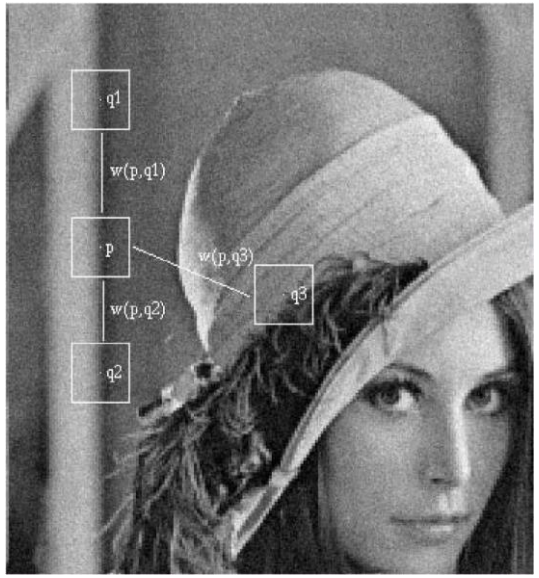
- NL-means(Non-Local means)
  - It estimates a target pixel as a weighted average of all the pixels in the image, with weights that depend on the similarity between them.

$$v'(i) = \sum_{j \in I} w(i,j) v(j)$$

  - For $w(i,j)$, it compares similarity between local patches centered at i and j, making it more robust.
  - It can now consider geometrical configuration in a whole neighborhood.

$$w(i,j) = \frac{1}{C} e^{-\frac{\left\| v(N_i) - v(N_j) \right\|_2^2}{h^2}},$$

where N_i is a vector of neighbors of i and $C = \sum_j e^{-\frac{\left\| v(N_i) - v(N_j) \right\|_2^2}{h^2}}$



(2005, Buades, Coll, Morel, A non-local algorithm for image denoising)

Classical Methods - NL-means

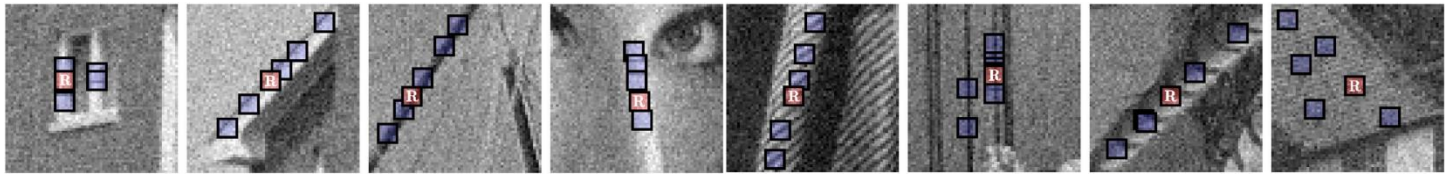
- Comparison between NL-means and previous methods

(2005, Buades, Coll, Morel, A non-local algorithm for image denoising)

Classical Methods - BM3D

- BM3D(Block-Matchig and 3D filtering)
  - Alike NL-means, it leverages NSS(Non-local Self-Similarity)
  - Instead of averaging, it does collaborative filtering so that it can perform better for natural images

  - Find d similar image patches with Block Matching, Grouping them into a 3D block

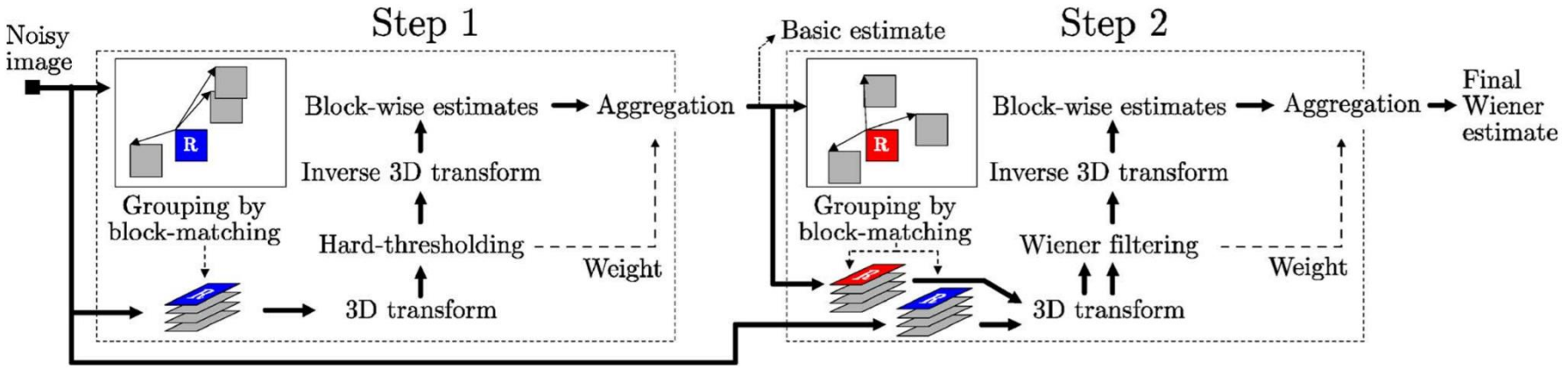


(2007, K. Dabov, et al., 'Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering')

  - Apply collaborative filtering in 3D-transform domain. It is similar to wavelet shrinkage
    1. apply $d+1$ dimensional linear transform
    2. shrink the transform coefficients (shrinkage in 3D transform domain)
    3. invert linear transform
  - Can achieve sparse representation of the true signal so that the noise can be well separated by shrinkage. (different from wavelet shrinkage for each separate patch)

Classical Methods - BM3D

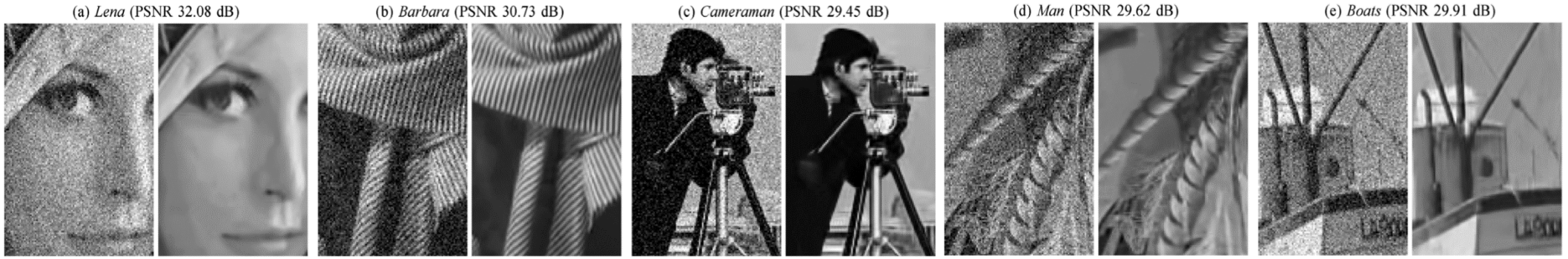
- It consists of 2 steps
  - In step 2, it takes basic estimation from step 1 as its input.
  - Then it applies 3D collaborative filtering to the total block, formed by merging blocks from each step



(2007, K. Dabov, et al., 'Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering')

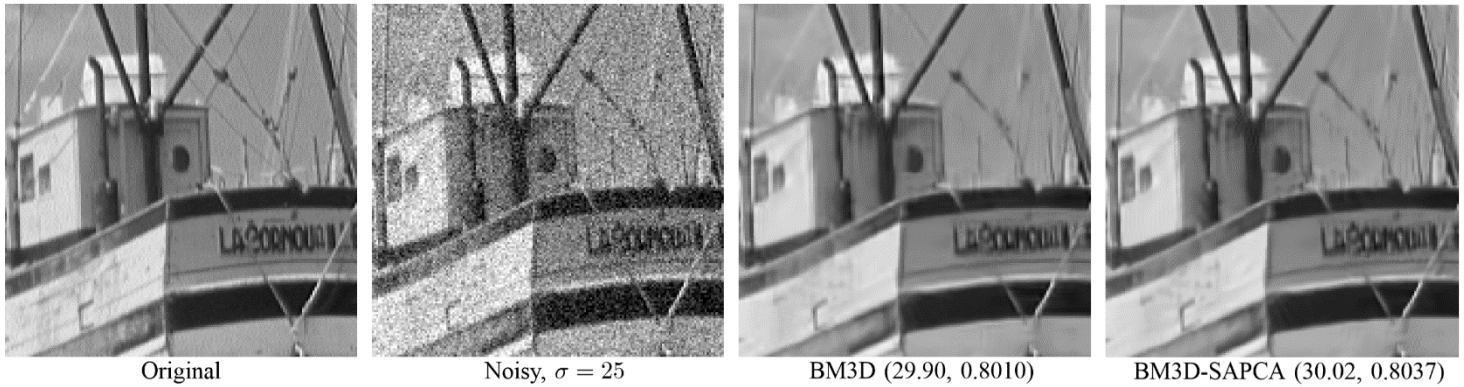
Classical Methods - BM3D

- Results



(2007, K. Dabov, et al., 'Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering')

- Some extensions were proposed by modifying its filtering or matching.



(2009, K. Dabov, et al., 'BM3D Image Denoising with Shape-Adaptive Principal Component Analysis')

Deep Denoising - Early Work

- As deep learning methods got attention, there was a move to apply it to denoising problem as well.

- DnCNN

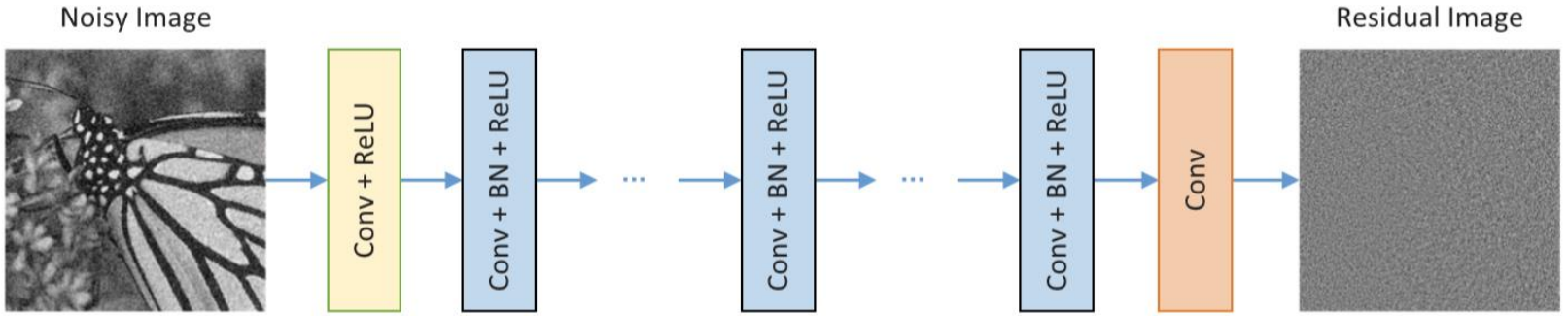
Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising

Kai Zhang, Wangmeng Zuo, *Senior Member, IEEE*, Yunjin Chen, Deyu Meng, and Lei Zhang, *Senior Member, IEEE*

  - One of the most important early deep-learning based denoising works.
  - Proposed a basic CNN architecture for denoising.
  - Became the basic comparison of other studies.

Deep Denoising - Early Work: DnCNN

- CNN architecture
  - Consists of ResBlocks (17 for non-blind setting, 20 for blind setting), with one layer of convolution for its head and tail each.



- Residual learning
  - It doesn't output estimated clear image directly
  - Instead, it outputs estimated noise map.
  - The desired cleared image is obtained by subtracting the noise map from the input.

(2017, K. Zhang, et al., 'Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising')

Deep Denoising - Early Work: DnCNN

- Learning scheme
  - Supervised learning with synthetic AWGN(Additive White Gaussian Noise)
  - Randomly cropped certain number of patches from the training dataset, and with additive noise, used them as training batches.

- Dataset: BSD (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/)
  - Originally introduced for segmentation task, but for its images were relatively clear, has been also used for denoising.

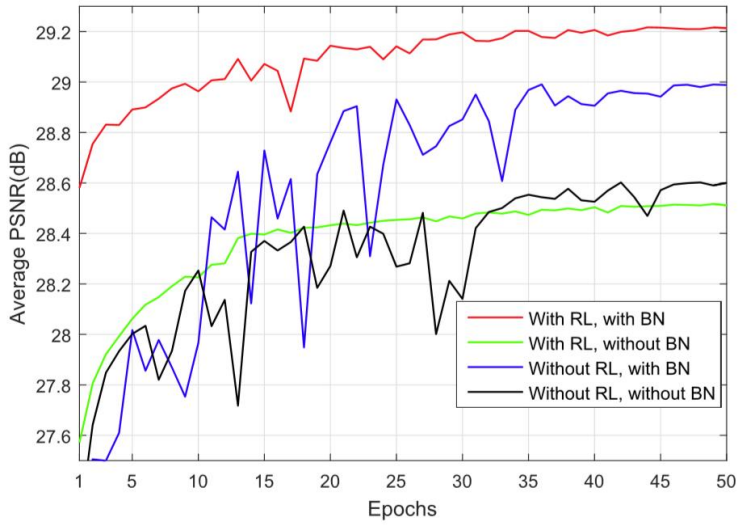


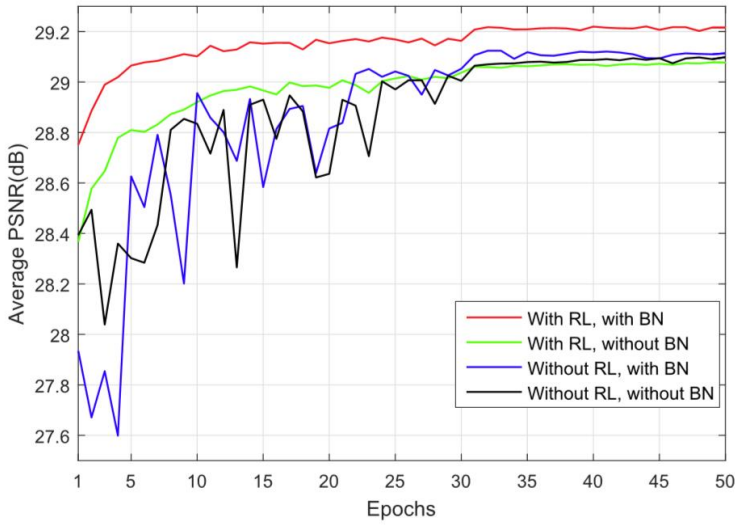


  - Followed its usage of previous works:
    - Used 400 cropped images for training of grayscale denoising model
    - Used 432 original images for training of color one
    - And used remaining 68 images for evaluation (disjoint with training set), with other test sets.

Deep Denoising - Early Work: DnCNN

- Effect of residual learning
  - Insists that residual learning makes it better and stable, especially when it is used with batch normalization.





(2017, K. Zhang, et al., 'Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising')

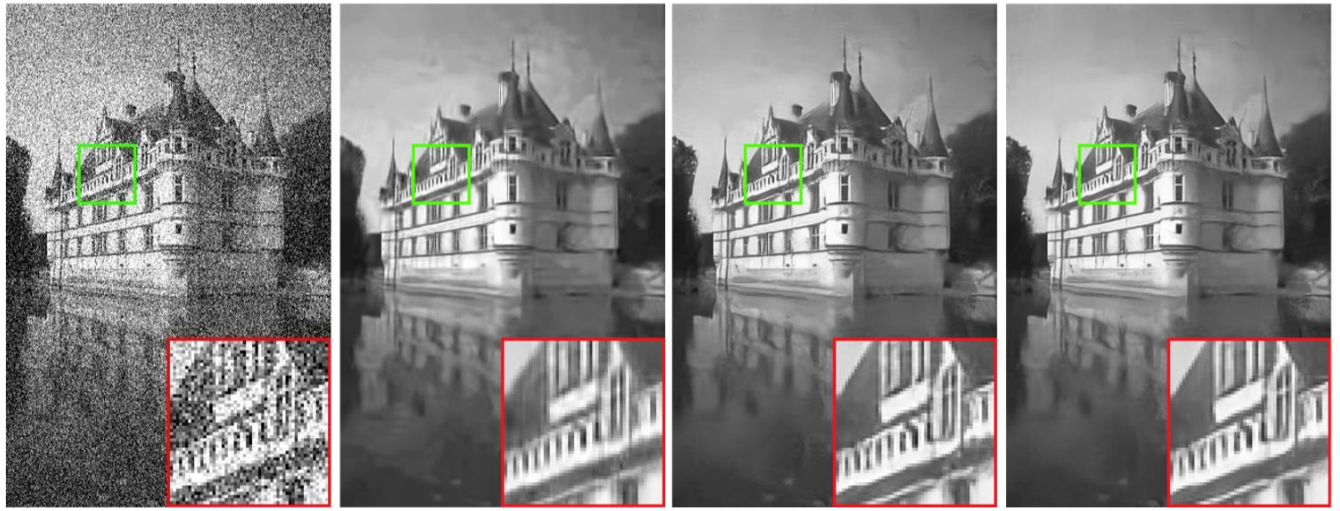
Deep Denoising - Early Work: DnCNN

- Got better results for both non-blind and blind model
  - DnCNN-S is for non-blind case, DnCNN-B for blind one.

THE AVERAGE PSNR(dB) RESULTS OF DIFFERENT METHODS ON THE BSD68 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN-S	DnCNN-B
$\sigma = 15$	31.07	31.37	31.21	-	31.24	31.42	**31.73**	31.61
$\sigma = 25$	28.57	28.83	28.68	28.96	28.74	28.92	**29.23**	29.16
$\sigma = 50$	25.62	25.87	25.67	26.03	-	25.97	**26.23**	**26.23**

  - The performance gap between blind and non-blind is not significant



noisy input | BM3D | DNCNN-S | DNCNN-B

(2017, K. Zhang, et al., 'Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising')

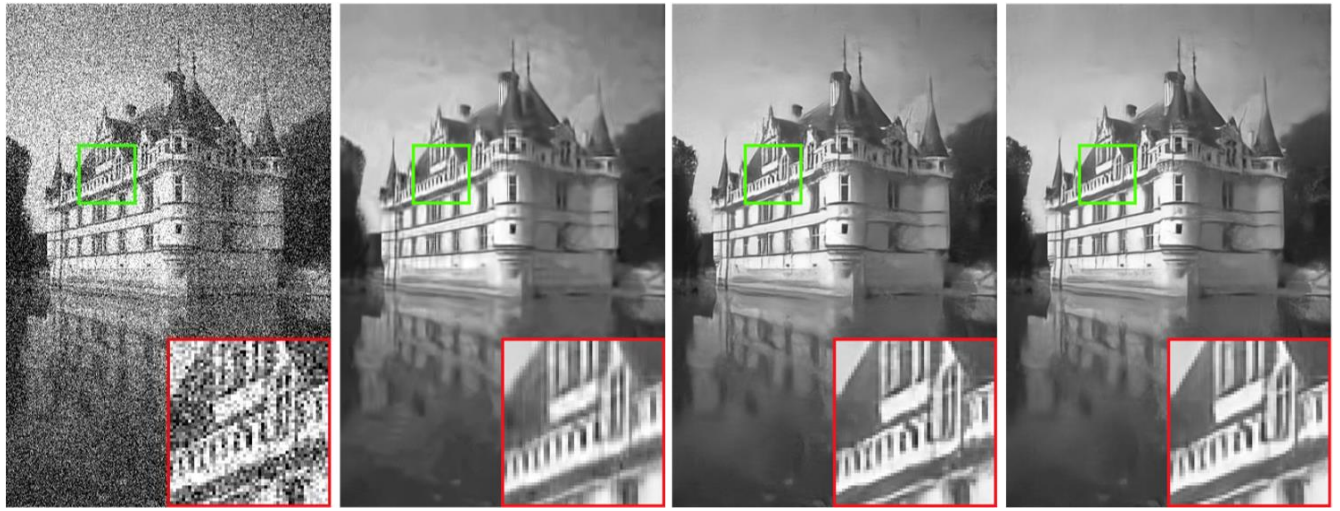
Deep Denoising - Early Work: DnCNN

- Got better results for both non-blind and blind model
  - DnCNN-S is for non-blind case, DnCNN-B for blind one.

THE AVERAGE PSNR(dB) RESULTS OF DIFFERENT METHODS ON THE BSD68 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN-S	DnCNN-B
$\sigma = 15$	31.07	31.37	31.21	-	31.24	31.42	**31.73**	31.61
$\sigma = 25$	28.57	28.83	28.68	28.96	28.74	28.92	**29.23**	29.16
$\sigma = 50$	25.62	25.87	25.67	26.03	-	25.97	**26.23**	**26.23**

  - The performance gap between blind and non-blind is not significant

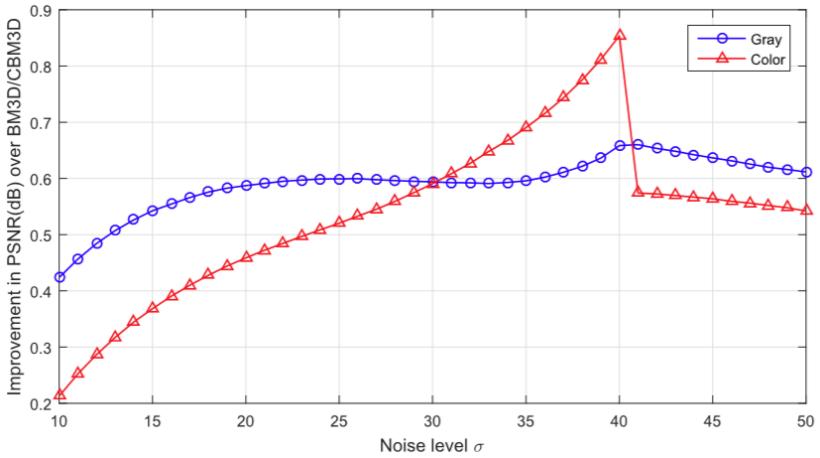


noisy input | BM3D | DNCNN-S | DNCNN-B

(2017, K. Zhang, et al., 'Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising')

Deep Denoising - Early Work: DnCNN

- Its blind model performs better in whole noise level range.



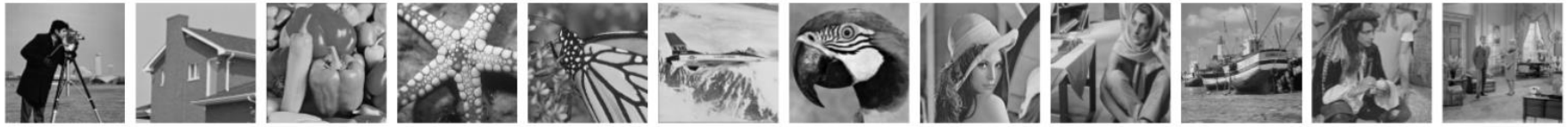
- It can also handle real-world image denoising reasonably.

noisy input | DnCNN-B | noisy input | DnCNN-B

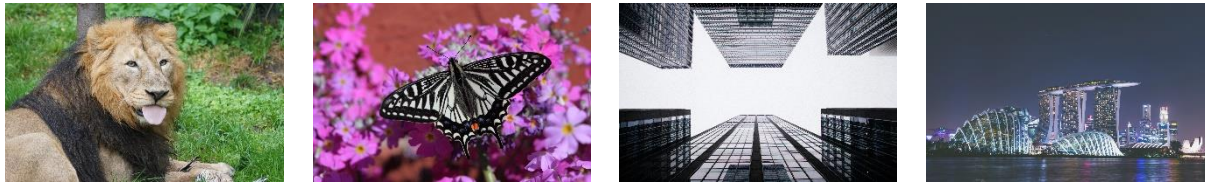
(2017, K. Zhang, et al., 'Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising')

Dataset and Benchmarks for Deep Denoising

- For synthetic noise: Need clear images
  - BSD: Used for early deep denoising works and became basic benchmark.
    (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/)
  - Set12, Classic5, LIVE1, Kodak: Famous traditional benchmarks used for image processing.



  - DIV2K: Introduced for NTIRE 2017, 2018 super-resolution challenge.
    HR images are used for denoising. (https://data.vision.ee.ethz.ch/cvl/DIV2K/)

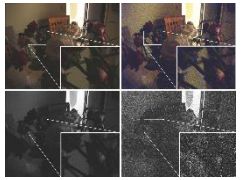


  - Urban100: Usually used as benchmark. Rich recurring pattern.

  - Cropped ImageNet: Sometimes used for training.
    Can use other information like segmentation map

Dataset and Benchmarks for Deep Denoising

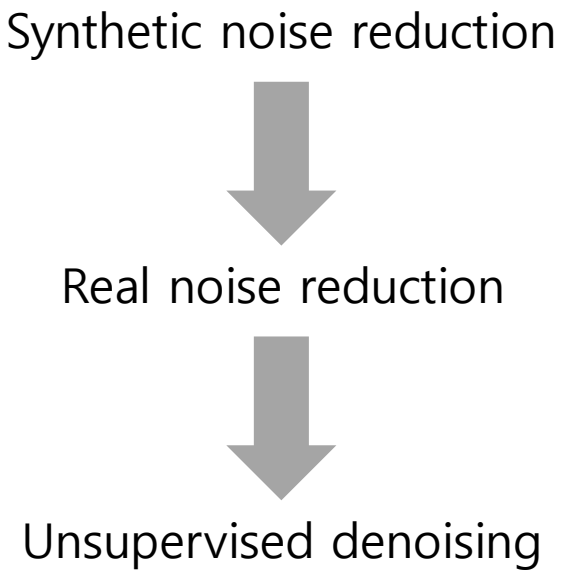
- For real noise: Need real noisy image from camera paired with clear one.
  - Clear images are generated by processing sequence of images from identical scene, with different camera setting.

  - SIDD: Noisy images from smartphone camera, and estimated ground truth clear images.
    Train and benchmark dataset for NTIRE 2019 Real Image Denoising Challenge.
    (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/)
    
    
  - DND: Image pairs captured with changing ISO and exposure. Benchmark GT is not open.
    (https://noise.visinf.tu-darmstadt.de/)
    
  - Renoir: Image pairs captured with changing ISO and exposure.
    (http://ani.stat.fsu.edu/~abarbu/Renoir.html)
    
  - Nam: 11 scenes averaged each 500 static images.

Deep Denoising – Important Works

- Now introduce some important works in deep denoising.

- Here handles mainly for **single image denoising**, in flow of



- And more about **video denoising**.

DDFN

Deep Boosting for Image Denoising

Chang Chen, Zhiwei Xiong(✉), Xinmei Tian, and Feng Wu

University of Science and Technology of China
changc@mail.ustc.edu.cn, {zwxiong,xinmei,fengwu}@ustc.edu.cn

- Improvement in architecture
  - Iterative usage of modules, in concept of Boosting
  - Dense connection and dilated convolution

(2018, C. Chen, et al., 'Deep Boosting for Image Denoising')

DDFN

- Basic idea of Boosting
  - Set of weak learners can work as a one strong learner, in perspective of general ML.

  - For deep denoising, applied it as iteratively extracting unrecovered signal from the residual and adding it back.

$$\hat{x}^{n+1} = F(y + \hat{x}^n) - \hat{x}^n$$

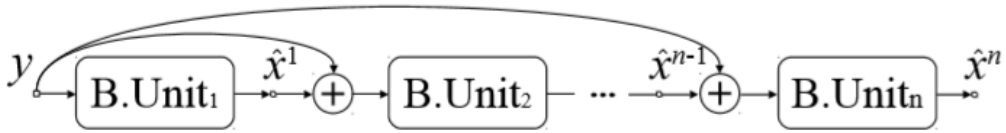
    - y is input measurement, $\hat{x}^n$ is estimated reconstruction in each step.
    - Here they chose $y + \hat{x}^n$ instead of residual

$$\begin{aligned} y + \hat{x} &= (x + u) + (x + v) \\ &= 2x + (u + v) \end{aligned}$$

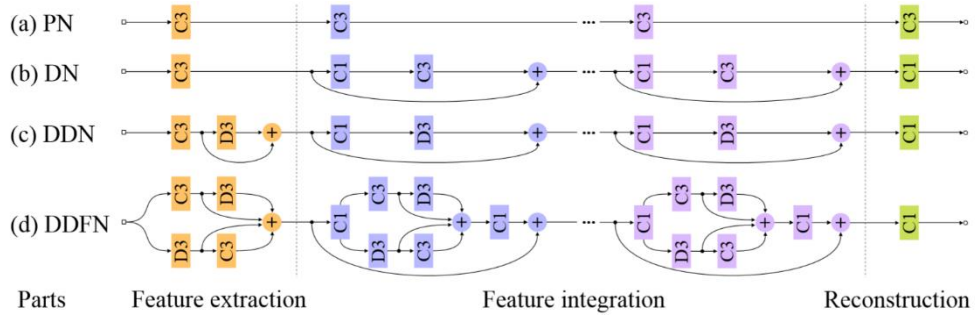
    meaning that its desired signal x is strengthened (in manner of SNR, by Cauchy-Schwarz ineq.)

DDFN

- Deep Boosting Architecture
  - This concept of boosting leads to following architecture design



  - One boosting unit is :



Parts	PN	DN	DDN	DDFN
1	$C(3\times3\times24)$	$C(3\times3\times32)$	$\begin{bmatrix}C(3\times3\times16)\\D(3\times3\times16)\end{bmatrix}$	$\begin{bmatrix}C(3\times3\times8/6)\\D(3\times3\times8/6)\end{bmatrix}$
2	$C(3\times3\times24)\times8$	$\begin{bmatrix}C(1\times1\times32)\\C(3\times3\times8)\end{bmatrix}\times8$	$\begin{bmatrix}C(1\times1\times32)\\D(3\times3\times8)\end{bmatrix}\times8$	$\left\{\begin{bmatrix}C(1\times1\times24)\\C(3\times3\times6/3)\\D(3\times3\times6/3)\\C(1\times1\times8)\end{bmatrix}\right\}\times8$
3	$C(3\times3\times1)$	$C(1\times1\times1)$	$C(1\times1\times1)$	$C(1\times1\times1)$

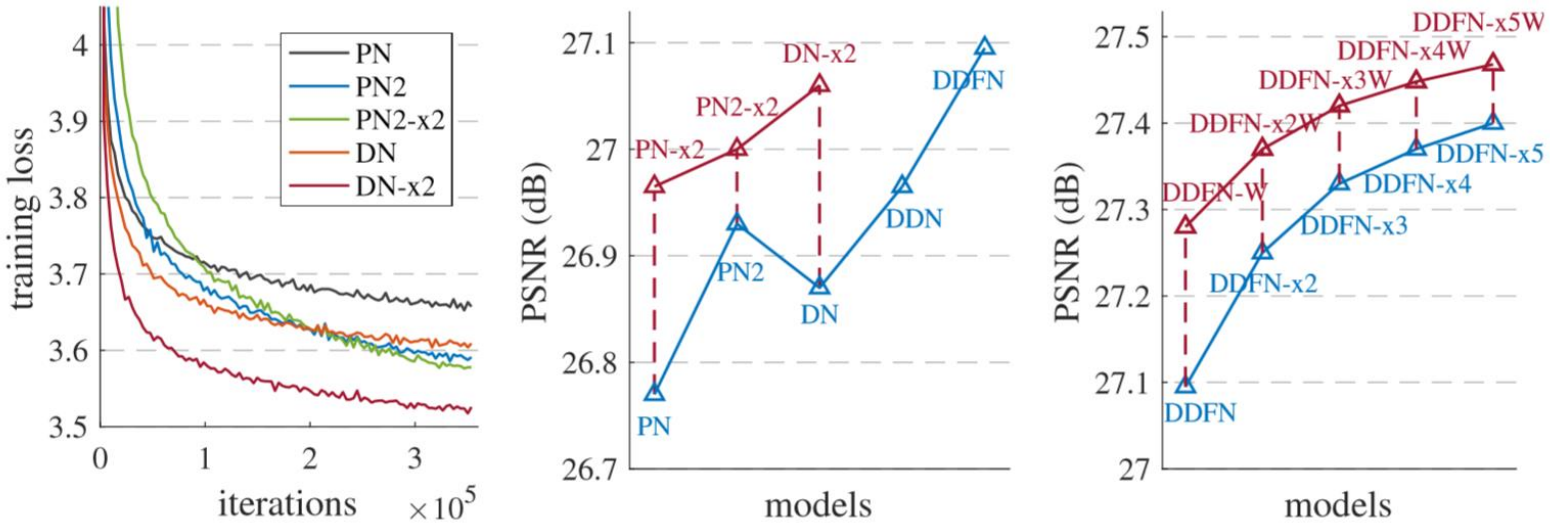
    - From up to bottom shows improvement in B.Unit design.

    - The ‘feature integration’ part is repeated 8 times, with dense connections between them.
    - Then, this whole B.Unit is repeated(5 for BSD68 experiment) also with dense connections.

(2018, C. Chen, et al., ‘Deep Boosting for Image Denoising’)

DDFN

- Deep Boosting Architecture
  - This concept of boosting leads to following architecture design



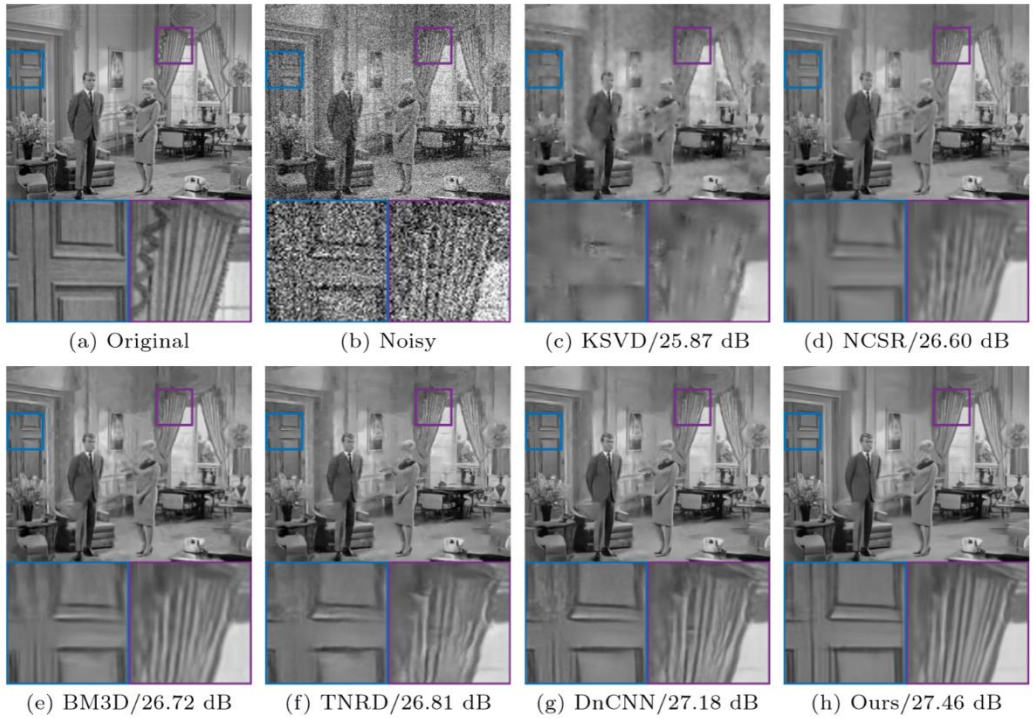
  - Dense connections inside boosting unit make convergence better.
  - Improvement in boosting unit leads to better performance
  - Iterative usage of DDFN has performance gain, can choose between performance-computation trade-off

(2018, C. Chen, et al., 'Deep Boosting for Image Denoising')

DDFN

- Non-blind denoising results on BSD68, Set12

Dataset	σ	MLP [10]	CSF [8]	GCRF [33]	TNRD [11]	NLNet [34]	DeepAM [35]	DnCNN [12]	DDFN/-x5W
Set12	15	-	32.32	-	32.50	-	-	32.86	32.73 **32.98**
	25	30.03	29.84	-	30.06	-	30.40	30.44	30.33 **30.60**
	50	26.78	-	-	26.81	-	-	27.18	27.10 **27.46**
BSD68	15	-	31.24	31.43	31.42	31.52	31.68	31.73	31.66 **31.83**
	25	28.96	28.74	28.89	28.92	29.03	29.21	29.23	29.16 **29.35**
	50	26.03	-	-	25.97	26.07	26.24	26.23	26.19 **26.42**



(a) Original (b) Noisy (c) KSVD/25.87 dB (d) NCSR/26.60 dB

(e) BM3D/26.72 dB (f) TNRD/26.81 dB (g) DnCNN/27.18 dB (h) Ours/27.46 dB

Comparison on non-blind denoising performance for Set12, σ=50

(2018, C. Chen, et al., 'Deep Boosting for Image Denoising')

DDFN

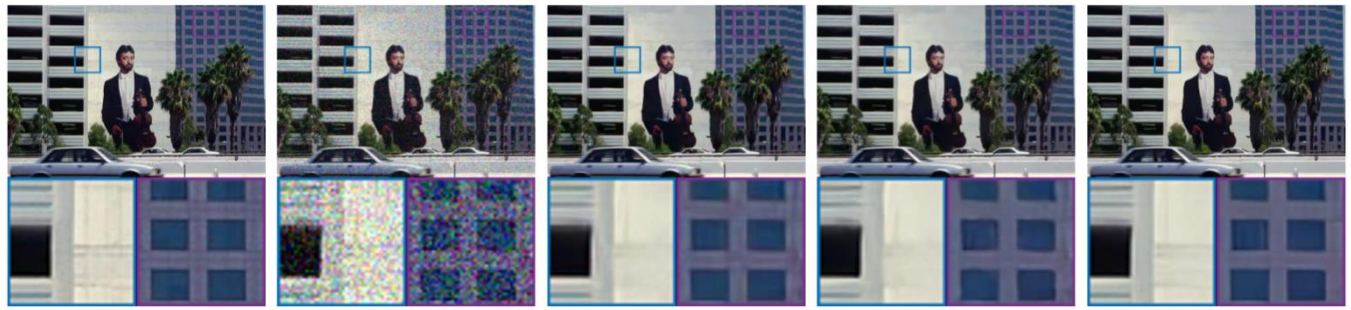
- Blind denoising results on CBSD68, Set12

for CBSD68

Models / σ	5	10	15	20	25	30	35	40	45	50
CBM3D [36]	40.24	35.91	33.52	31.91	30.71	29.73	28.89	28.09	27.84	27.38
DnCNN-B [12]	40.10	36.12	33.89	32.37	31.23	30.32	29.58	28.95	28.40	27.92
DDFN-x3W-B	**40.47**	**36.42**	**34.17**	**32.65**	**31.52**	**30.62**	**29.88**	**29.26**	**28.72**	**28.26**

for Set12

Models / σ	5	10	15	20	25	30	35	40	45	50
BM3D [7]	38.03	34.38	32.37	31.01	29.97	29.13	28.40	27.65	27.19	26.72
DnCNN-B [12]	37.70	34.53	32.68	31.38	30.36	29.53	28.83	28.22	27.69	27.21
DDFN-B	37.31	34.30	32.45	31.15	30.13	29.30	28.60	27.98	27.44	26.96
DDFN-x5W-B	**38.14**	**34.76**	**32.86**	**31.55**	**30.54**	**29.71**	**29.02**	**28.41**	**27.87**	**27.44**



(a) Original (b) Noisy (c) CBM3D/28.86 dB (d) CDnCNN/29.46 dB (e) Ours/29.92 dB

Comparison on blind denoising performance for CBSD68, σ=35

(2018, C. Chen, et al., 'Deep Boosting for Image Denoising')

CBDNet

Toward Convolutional Blind Denoising of Real Photographs

Shi Guo[1,3,4], Zifei Yan(✉) [1], Kai Zhang[1,3], Wangmeng Zuo[1,2], Lei Zhang[3,4]
[1]Harbin Institute of Technology, Harbin; [2]Peng Cheng Laboratory, Shenzhen;
[3] The Hong Kong Polytechnic University, Hong Kong; [4]DAMO Academy, Alibaba Group
guoshi28@outlook.com, {wmzuo,yanzifei}@hit.edu.cn
cskaizhang@gmail.com, cslzhang@comp.polyu.edu.hk

- Intermediate noise level estimation towards real image denoising
  - Previous deep denoisers' performance collapses when applied to real noisy images.
  - Got intermediate noise map estimation, with corresponding loss terms.
  - Estimated noise map is used as a guide to a denoiser.

(2018, S. Guo, et al., 'Toward Convolutional Blind Denoising of Real Photographs')

CBDNet

- Generating synthetic real noisy image
  - With real noisy images, also used generated noise map for training.
    - Noise by photon sensing -> known to be approximated as Poisson noise, n_s
      Its noise level is proportional to signal irradiance
    - Stationary signal-independent noise -> approximated as Gaussian noise, n_c
    - Total generated noise term is:

$$n(L) = n_s(L) + n_c,$$

$$\mathbf{L} \cdot \sigma_s^2 \qquad \sigma_c^2$$

(L: irradiance of each pixel)

    - Added this term to raw image L, applied camera function to get synthetic real noisy image

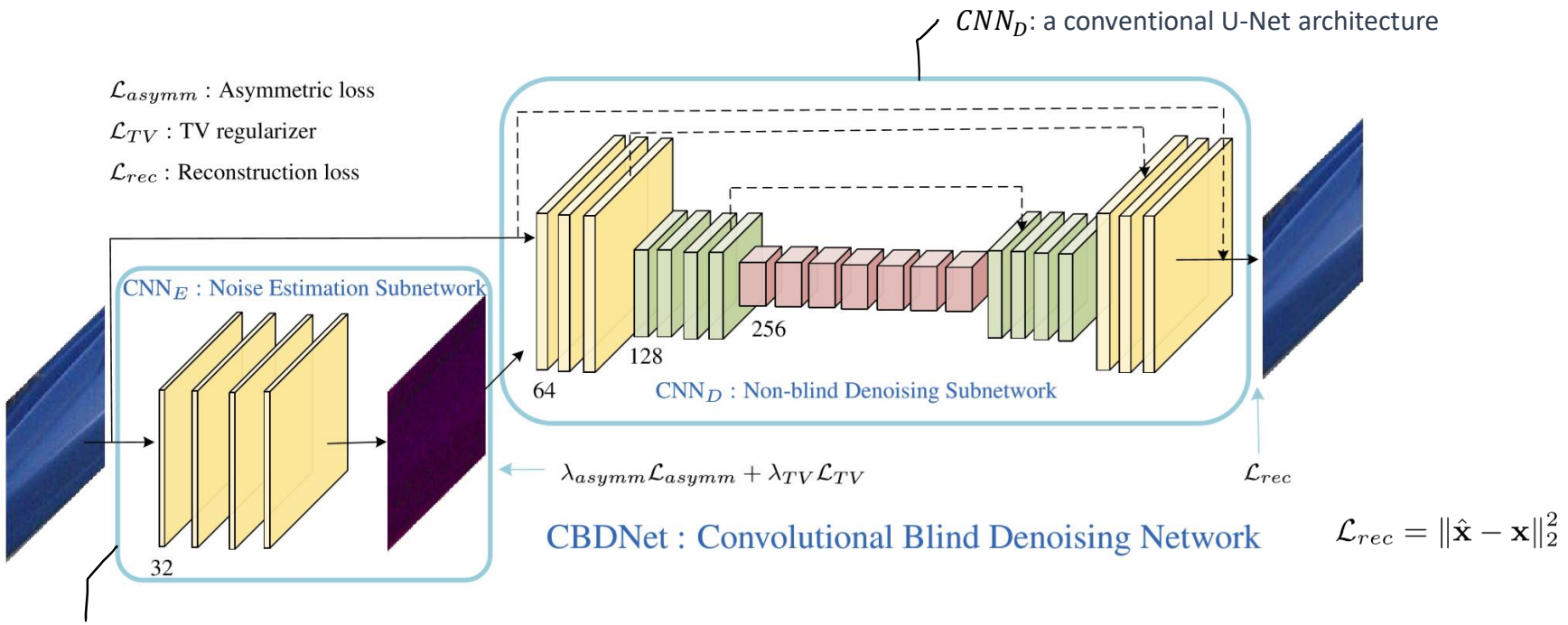
$$\mathbf{y} = f(\mathbf{DM}(\mathbf{L} + \mathbf{n}(\mathbf{L}))) \qquad \mathbf{L} = \mathbf{M}f^{-1}(\mathbf{x})$$

f: camera response function, random-sampled from 201 CRFs in previous work
DM: demosaicing, interpolation function adopted from other work
M: from BGR image to Bayer image

(2018, S. Guo, et al., 'Toward Convolutional Blind Denoising of Real Photographs')

CBDNet

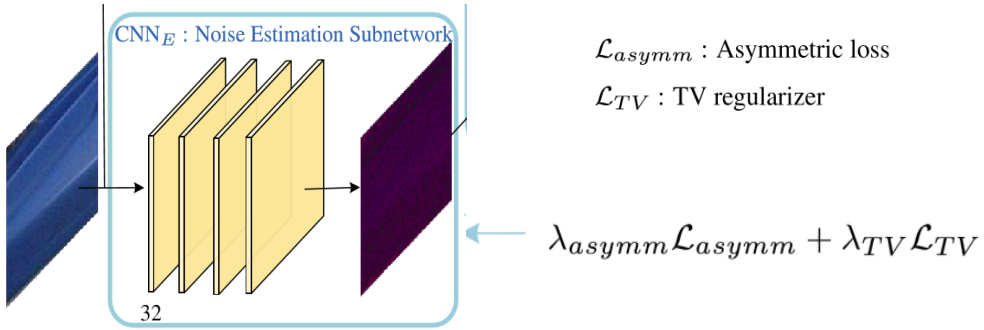
- Total network consists of two parts.
  - CNN_E, a noise level(total σ) estimation subnetwork
  - CNN_D, a (as to say) non-blind denoising subnetwork



(2018, S. Guo, et al., 'Toward Convolutional Blind Denoising of Real Photographs')

CBDNet

- Giving supervision to noise estimation
  - when using synthetic noisy images, give supervision with its generated noise level map



  - "Assymetric": performance drop is bigger when noise level is under-estimated than over-estimated.
    (2017, K. Zhang, et al., 'FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising)
  - Assymetric is given as

$$L_{asymm} = \sum_i \left|\alpha - \mathrm{I}_{(\hat{\sigma}(y_i)-\sigma(y_i))<0}\right| \cdot (\hat{\sigma}(y_i) - \sigma(y_i))^2,$$

where $\alpha < 0.5$ (typically 0.3~0.5)

  - TV regularizer is to restrain it to be smooth enough

$$\mathcal{L}_{TV} = \|\nabla_h \hat{\sigma}(\mathbf{y})\|_2^2 + \|\nabla_v \hat{\sigma}(\mathbf{y})\|_2^2$$

(2018, S. Guo, et al., 'Toward Convolutional Blind Denoising of Real Photographs')

CBDNet

- Training data
  - For synthetic images, used BSD, Waterloo, MIT-Adobe-FiveK dataset.

    (2017, M, Kede, et al., 'Waterloo Exporation Database: New Challenges for Image Quality Assessment Models')

    (2011, V. Bychkovsky, et al., 'Learning Photographic Global Tonal Adjustment with a Database of Input / Output Image Pairs')
  - If the input is synthetic, gave supervision to the estimated noise map.
  - For real noisy images, used RENOIR dataset.

- Test data
  - DND
  - Nam (ramdomly cropped 512*512)
  - NC12 (GT not available, just for qualitative evaluation)

CBDNet

- Results

Table 1: The quantitative results on the DND benchmark.

Method	Blind/Non-blind	Denoising on	PSNR	SSIM
CDnCNN-B [61]	Blind	sRGB	32.43	0.7900
EPLL [64]	Non-blind	sRGB	33.51	0.8244
TNRD [11]	Non-blind	sRGB	33.65	0.8306
NCSR [13]	Non-blind	sRGB	34.05	0.8351
MLP [6]	Non-blind	sRGB	34.23	0.8331
FFDNet [62]	Non-blind	sRGB	34.40	0.8474
BM3D [12]	Non-blind	sRGB	34.51	0.8507
FoE [52]	Non-blind	sRGB	34.62	0.8845
WNNM [17]	Non-blind	sRGB	34.67	0.8646
GCBD [10]	Blind	sRGB	35.58	0.9217
CIMM [5]	Non-blind	sRGB	36.04	0.9136
KSVD [3]	Non-blind	sRGB	36.49	0.8978
MCWNNM [59].	Blind	sRGB	37.38	0.9294
TWSC [58]	Blind	sRGB	37.94	0.9403
CBDNet(Syn)	Blind	sRGB	37.57	0.9360
CBDNet(Real)	Blind	sRGB	37.72	0.9408
CBDNet(All)	Blind	sRGB	**38.06**	**0.9421**

Table 2: The quantitative results on the Nam dataset [43].

Method	Blind/Non-blind	PSNR	SSIM
NI [2]	Blind	31.52	0.9466
CDnCNN-B [61]	Blind	37.49	0.9272
TWSC [58]	Blind	37.52	0.9292
MCWNNM [59]	Blind	37.91	0.9322
BM3D [12]	Non-blind	39.84	0.9657
NC [29]	Blind	40.41	0.9731
WNNM [17]	Non-blind	41.04	0.9768
CBDNet	Blind	40.02	0.9687
CBDNet(JPEG)	Blind	**41.31**	**0.9784**

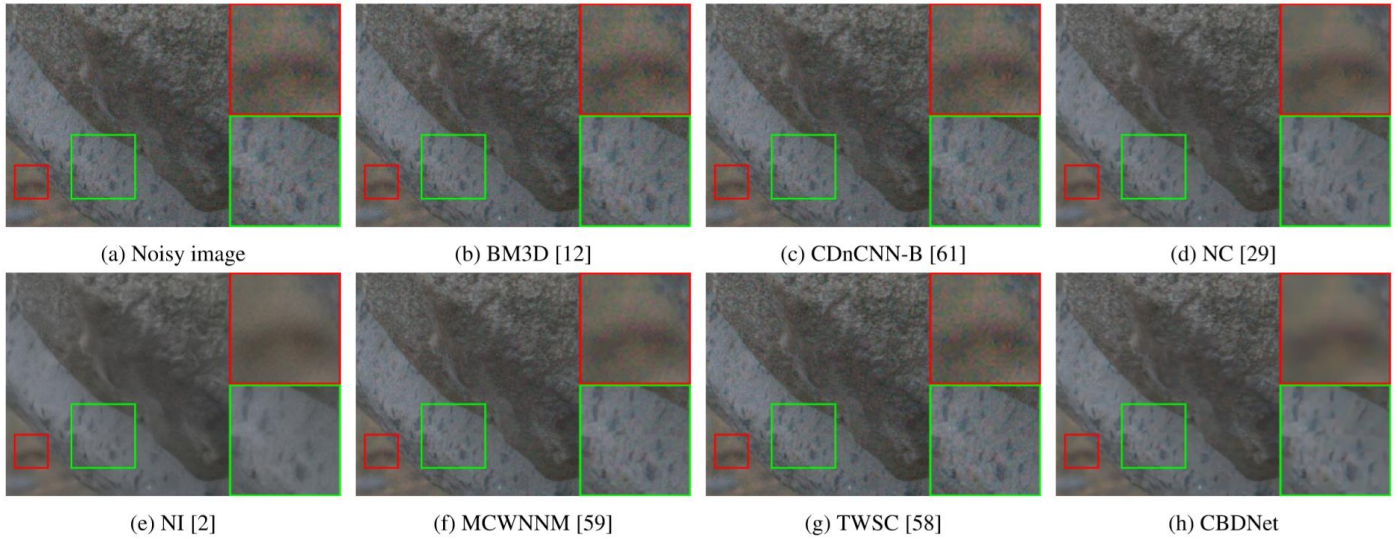
Table 3: PSNR/SSIM results by different noise models.

Method	DND [45]	Nam [43]
CBDNet(G)	32.52 / 0.79	37.62 / 0.9290
CBDNet(HG)	33.70 / 0.9084	38.40 / 0.9453
CBDNet(G+ISP)	37.41 / 0.9353	39.03 / 0.9563
CBDNet(HG+ISP)	**37.57 / 0.9360**	39.20 / 0.9579
CBDNet(JPEG)	—	**40.51 / 0.9745**

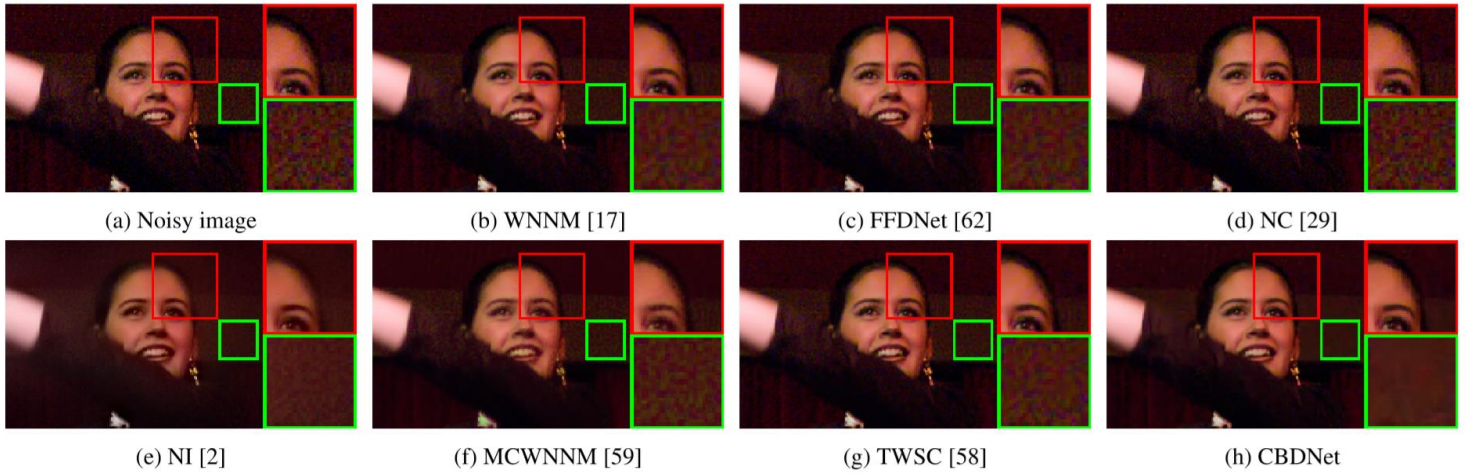
(2018, S. Guo, et al., 'Toward Convolutional Blind Denoising of Real Photographs')

CBDNet

- Results



Comparison on denoising performance for DND



Comparison on denoising performance for NC12

(2018, S. Guo, et al., 'Toward Convolutional Blind Denoising of Real Photographs')

Noise2Noise

Noise2Noise: Learning Image Restoration without Clean Data

Jaakko Lehtinen[1 2] Jacob Munkberg[1] Jon Hasselgren[1] Samuli Laine[1] Tero Karras[1] Miika Aittala[3] Timo Aila[1]

- Towards unsupervised learning for denoising
  - It is simply based on statistical characteristic, and works only for limited cases.
  - In fact it needs multiple noisy images. It's contribution is not for complete unsupervised learning, but for saving 'capture budget'.
  - It is used or modified in other works deal with unsupervised denoising.

Noise2Noise

- Basic idea
  - It is based on simple statistical characteristic of the noise.
  - For general estimating single value y from observations z,

$$\underset{z}{\operatorname{argmin}}\, \mathbb{E}_y\{L(z,\, y)\}$$

    leads to mean/median of z for L2/L1 loss,

  - For estimating clear image y from observations $\hat{x}_i$,
    using noisy $\hat{y}_i$ instead of correct y

$$\underset{\theta}{\operatorname{argmin}} \sum_i L\left(f_\theta(\hat{x}_i),\, \hat{y}_i\right)$$

    also leads to same mean/median of observations,
    if the noise is **zero-mean and independent from signal y**.
    (if not, the loss formulation should be changed)

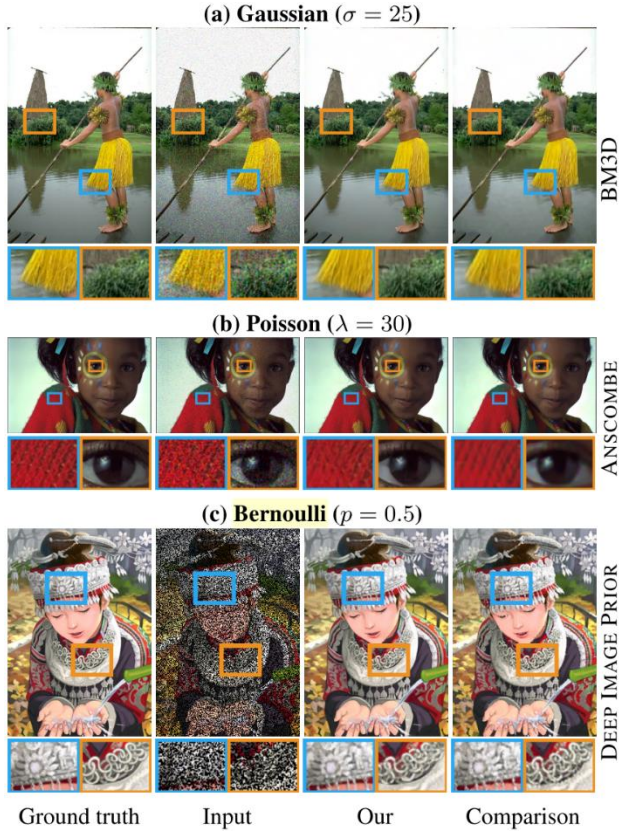
(2018, J. Lehtinen, et al., 'Noise2Noise: Learning Image Restoration without Clean Data')

Noise2Noise

- This means one can do same minimization with input/target drawn from **both corrupted distribution**
  - Performance on traditional benchmarks

	Gaussian (σ=25)			Poisson (λ=30)			Bernoulli (p=0.5)		
	clean	noisy	BM3D	clean	noisy	ANSC	clean	noisy	DIP
Kodak	32.50	32.48	31.82	31.52	31.50	29.15	33.01	33.17	30.78
BSD300	31.07	31.06	30.34	30.18	30.16	27.56	31.04	31.16	28.97
Set14	31.31	31.28	30.50	30.07	30.06	28.36	31.51	31.72	30.67
Average	**31.63**	**31.61**	**30.89**	**30.59**	**30.57**	**28.36**	**31.85**	**32.02**	**30.14**

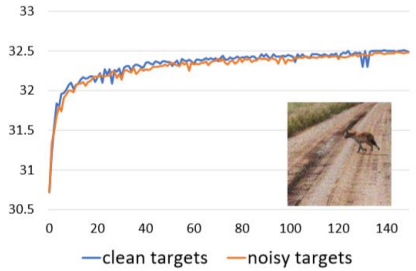
  - Its performance is comparable to that of using clean target
  - Shown better performance compared to previous GT-free methods



(2018, J. Lehtinen, et al., 'Noise2Noise: Learning Image Restoration without Clean Data')

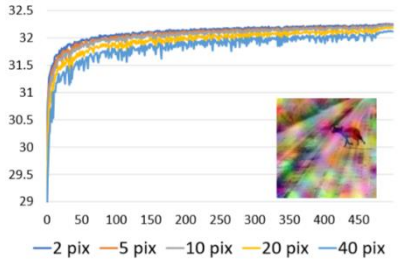
Noise2Noise

- Study for its convergence



(a) White Gaussian, $\sigma = 25$

  - (noisy -> noisy) setting seems impossible, as it's transforming one noise to another
  - However, the weight gradient is averaged over whole image pixels, making it possible (if noise is i.i.d)



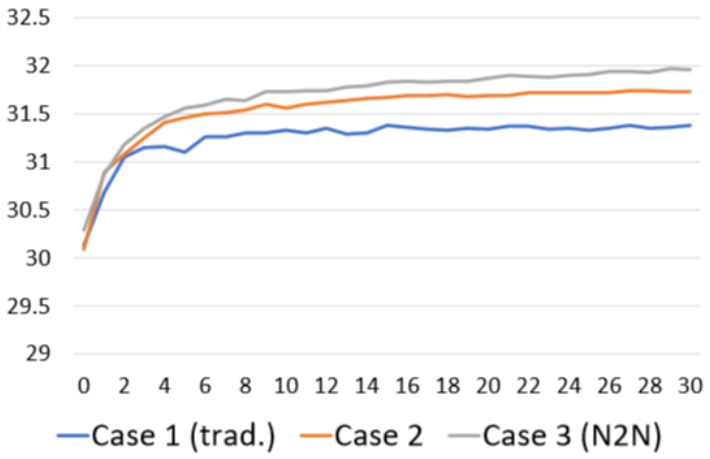
(b) Brown Gaussian, $\sigma = 25$

  - Slower convergence for brown Gaussian(normal noise blurred with Gaussian kernel), but eventually its quality is similar

(2018, J. Lehtinen, et al., 'Noise2Noise: Learning Image Restoration without Clean Data')

Noise2Noise

- Study for ‘Capture Budget’
  - ‘Capture Unit(CU)’: one image patch provided
  - Supposed 20 CUs are enough for one clean capture (for SIDD dataset, 150 successive images are used for this)
  - N: num. of clean latent
  - M: num. of noise realizations per clean latent



(c) Capture budget study (see text)

Case 1(traditional): N=100, M=20, total=2,000
Case 2: N=100, M=20, total=100*20*19=38,000
Case 3: N=1000, M=2 (preserving same CU)

=> Corrupted targets offer benefits

- Seeing more realization for same clean image
- Even with just two corrupted ones of each clean image

(2018, J. Lehtinen, et al., ‘Noise2Noise: Learning Image Restoration without Clean Data’)

Noise2Noise

- Results for various degradations
  - Text removal

For this, the color change caused by text is not zero-mean, so used L1 loss.

  - Random-valued impulse: harder version of impulse(salt-and-pepper) noise, for its values are random, instead of 0 and 1

For this, reduced γ value from 2 to 0, eventually to L0 loss from following term:

$$(|f_\theta(\hat{x}) - \hat{y}| + \epsilon)^\gamma$$

(2018, J. Lehtinen, et al., 'Noise2Noise: Learning Image Restoration without Clean Data')

GCBD, GAN2GAN, GRDN

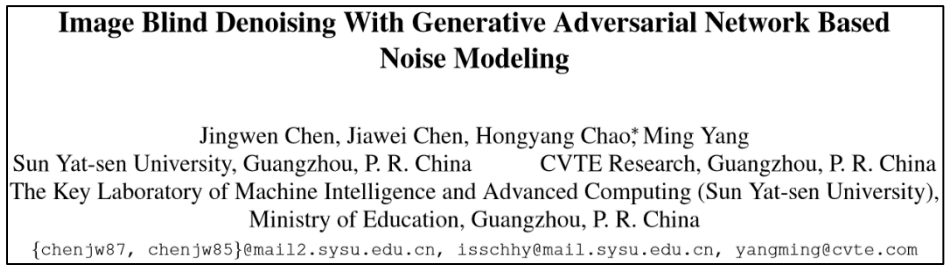
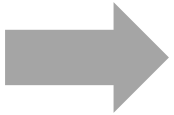
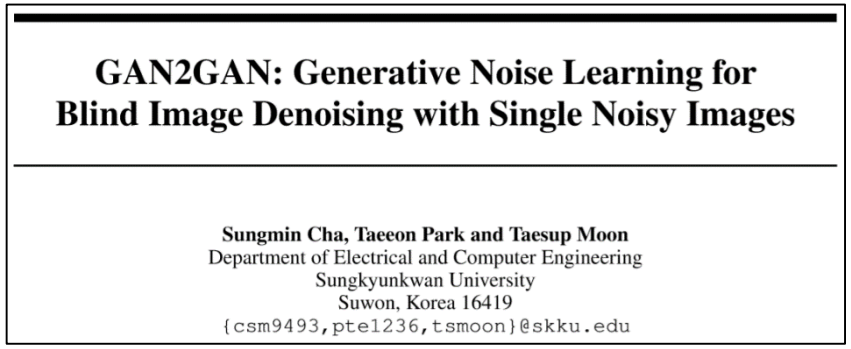


Image Blind Denoising With Generative Adversarial Network Based Noise Modeling

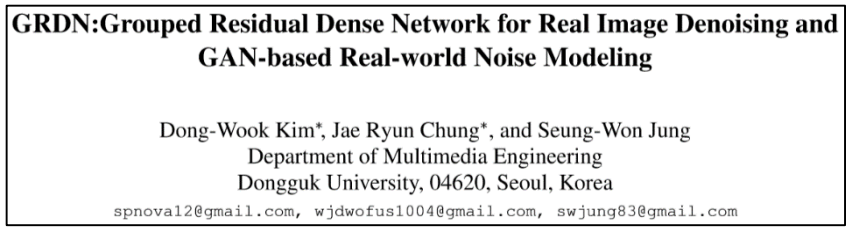
Jingwen Chen, Jiawei Chen, Hongyang Chao*, Ming Yang
Sun Yat-sen University, Guangzhou, P. R. China CVTE Research, Guangzhou, P. R. China
The Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, Guangzhou, P. R. China
{chenjw87, chenjw85}@mail2.sysu.edu.cn, isschhy@mail.sysu.edu.cn, yangming@cvte.com





GAN2GAN: Generative Noise Learning for Blind Image Denoising with Single Noisy Images

Sungmin Cha, Taeeon Park and Taesup Moon
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Korea 16419
{csm9493,pte1236,tsmoon}@skku.edu



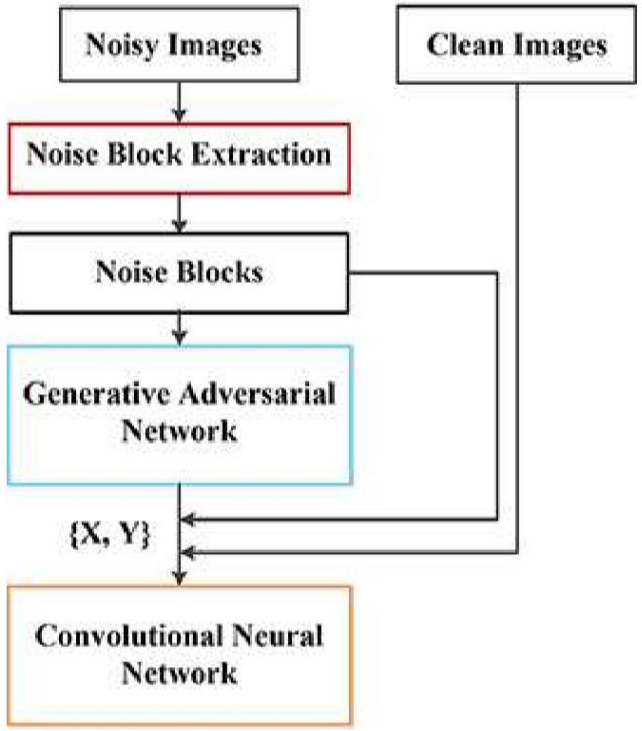
GRDN:Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling

Dong-Wook Kim*, Jae Ryun Chung*, and Seung-Won Jung
Department of Multimedia Engineering
Dongguk University, 04620, Seoul, Korea
spnova12@gmail.com, wjdwofus1004@gmail.com, swjung83@gmail.com

- Generating real noisy images for training
  - For there are no sufficient real noisy image data
  - Used GAN(Generative Adversarial Network)

GCBD

- Data pair is built during training, with trained GAN.
  - The GAN generates noise map, which will be added to clean images.



(2018, J. Chen, et al., 'Image Blind Denoising With Generative Adversarial Network Bsed Noise Modeling')

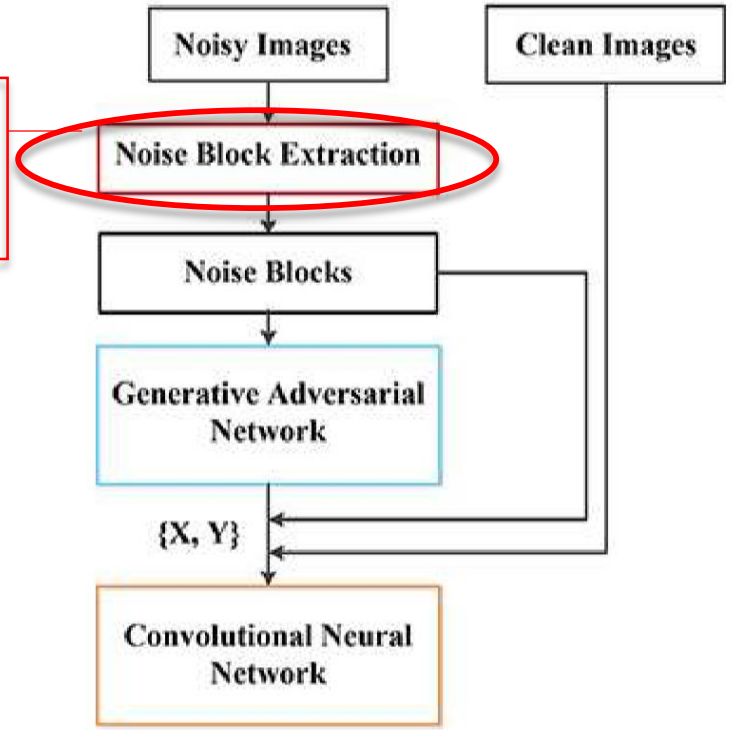
GCBD

- Noise block extraction for correct GAN training
  - Algorithm for searching smooth patches
  - For a patch, it examines its sub-patches to have similar means and variances.

For a patch p and its sub-path q, examine:

$$|Mean(\mathbf{q_j^i}) - Mean(\mathbf{p_i})| \leq \mu \cdot Mean(\mathbf{p_i})$$

$$|Var(\mathbf{q_j^i}) - Var(\mathbf{p_i})| \leq \gamma \cdot Var(\mathbf{p_i})$$

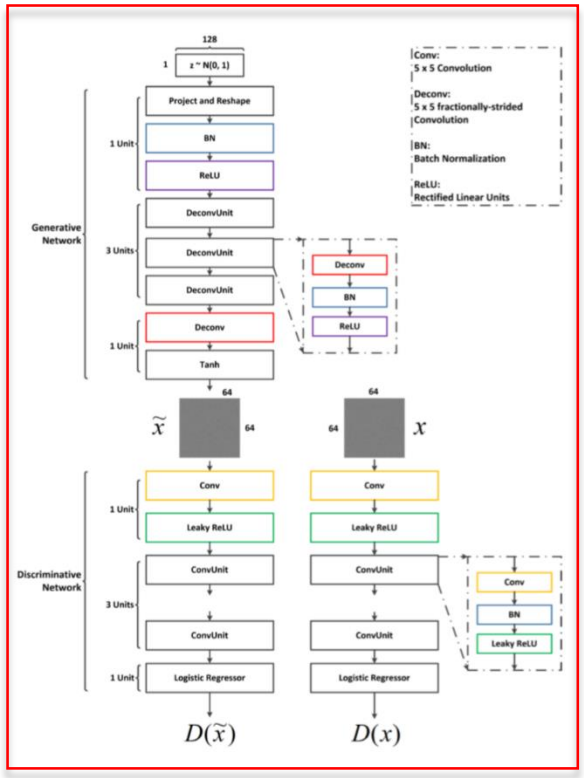


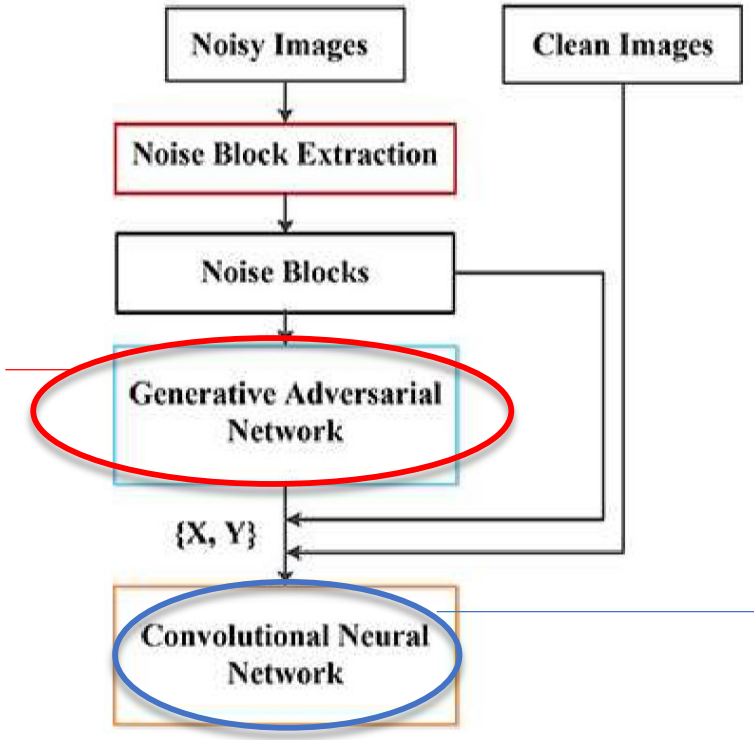
- The approximate noise block is derived as $v_i = s_i - Mean(s_i)$, where s_i is the patch found.

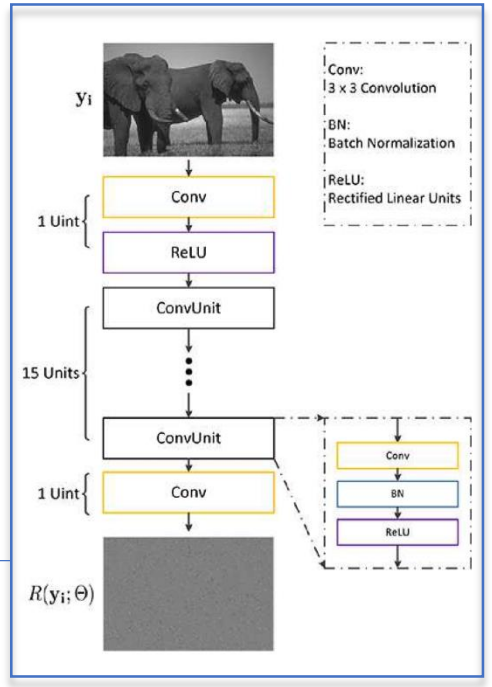
(2018, J. Chen, et al., 'Image Blind Denoising With Generative Adversarial Network Bsed Noise Modeling')

GCBD

- GAN and denoiser
  - For GAN, used wGAN-GP method.
  - For denoiser, used a simple CNN







(2018, J. Chen, et al., 'Image Blind Denoising With Generative Adversarial Network Bsed Noise Modeling')

GAN2GAN

- Improved smooth noisy patch extraction
  - Previous extraction rule of GCBD also picks patches with high-frequency repeating patterns, which is wrong.
  - With 2D DWT(Discrete Wavelet Transform) coefficients W_i of four sub-bands(i=1~4),

$$\frac{1}{4}\sum_{k=1}^{4}\left|\mathbf{Var}(W_k(\boldsymbol{p})) - \mathbb{E}[\mathbf{Var}_W(\boldsymbol{p})]\right| \leq \lambda\mathbb{E}[\mathbf{Var}_W(\boldsymbol{p})]$$

where $\mathbb{E}[\mathbf{Var}_W(\boldsymbol{p})] \triangleq \frac{1}{4}\sum_{k=1}^{4}\mathbf{Var}(W_k(\boldsymbol{p}))$

  - Means that if variance in transform space of each band is similar to the average of them.

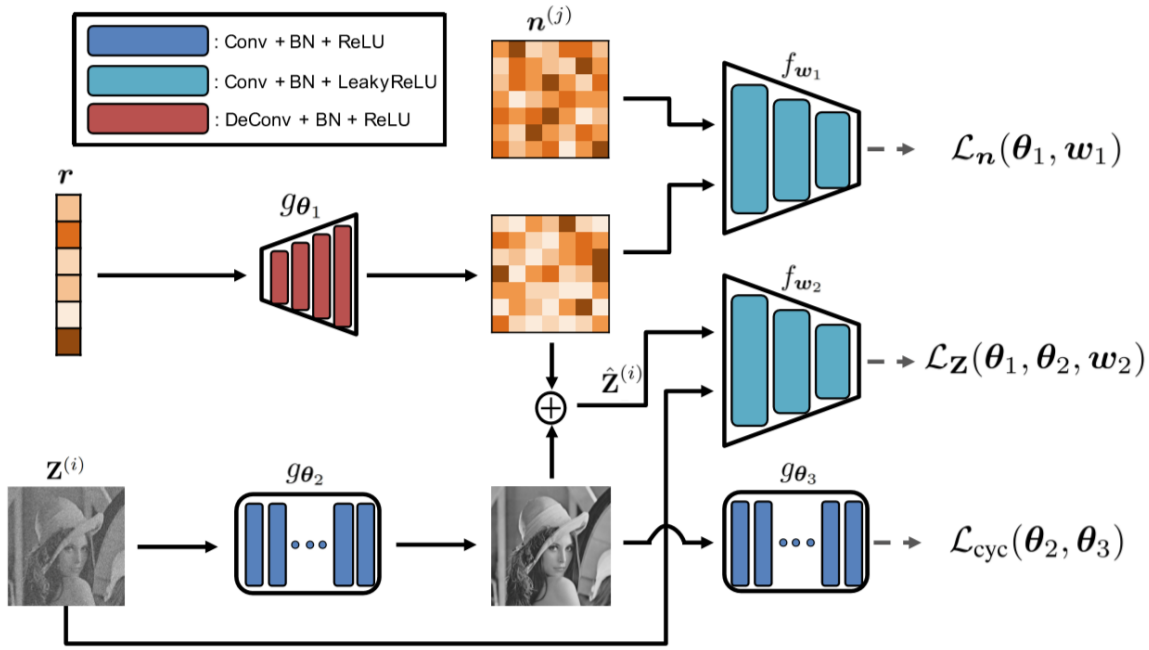
    (Check traditional wavelet transform based denoising, which limits highest number of DWT coefficients and take inverse transform)

  - Also it doesn't have to check all the sub-patches.

(2019, S. Cha, et al., 'GAN2GAN: Generative Noise Learning for Blind Image Denoising with Single Noisy Images')

GAN2GAN

- Improved GAN training
  - It also discriminates if the generated noisy image is realistic(L_z), not only noise blocks(L_n).
  - It is to give information to denoiser that this noise is independent from clear image.
  - Also with same(not shared) denoiser g_{θ_2} and g_{θ_3}, gave cyclic loss



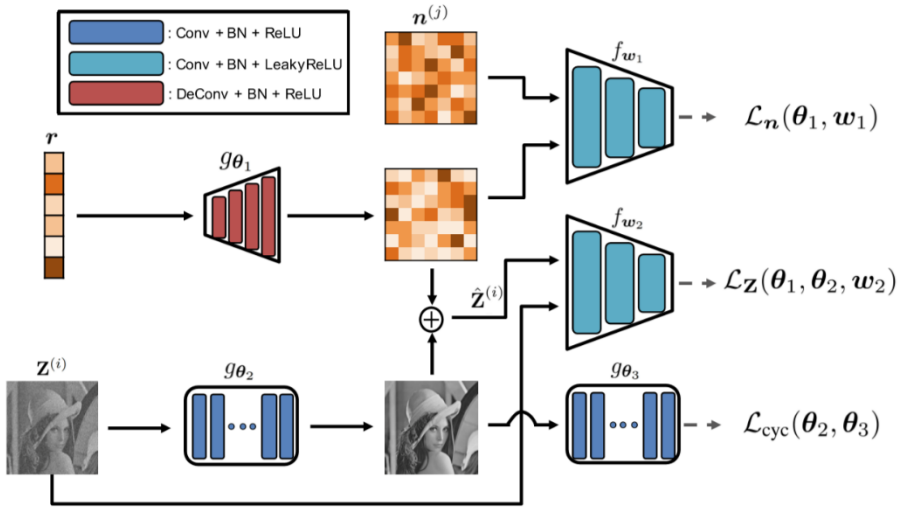
(2019, S. Cha, et al., 'GAN2GAN: Generative Noise Learning for Blind Image Denoising with Single Noisy Images')

GAN2GAN

- Iterative training
  - After training total GAN, can train another denoiser $\hat{X}_\theta(z)$ with N2N training,

$$\mathcal{L}_{\mathrm{N2N}}(\boldsymbol{\theta}, \hat{\mathcal{D}}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\mathbf{Z}}_1^{(i)} - \hat{X}_{\boldsymbol{\theta}}(\hat{\mathbf{Z}}_2^{(i)}) \right)^2$$

    simply using the generated noisy images, $\hat{Z}_2$
    (However, this basically assume the noise to be zero-mean)
  - And after that, can replace previous denoiser g_{θ_2} into $\hat{X}_\theta$, the upgraded denoiser.



(2019, S. Cha, et al., 'GAN2GAN: Generative Noise Learning for Blind Image Denoising with Single Noisy Images')

GAN2GAN

- Results
  - For AWGN on BSD68

PSNR /SSIM	BM3D	DnCNN-S	DnCNN-B	N2N	N2V	G2G (BSD)	$G2G_I$ (BSD)	G2G (LM,BSD)	$G2G_I$ (LM,BSD)
σ=15	31.07 /0.8717	31.54 /0.8848	31.14 /0.8689	31.22 /0.8793	28.32 /0.7883	30.98 /0.8552	31.51 **/0.8827**	31.27 /0.8700	**31.55** /0.8826
σ=25	28.56 /0.8013	29.03 /0.8167	28.84 /0.8132	28.82 /0.8132	26.69 /0.7093	28.23 /0.7669	28.82 0.8056	28.65 /0.7901	**28.93 0.8079**
σ=30	27.74 /0.7727	28.13 /0.7860	27.99 /0.7857	27.95 /0.7824	26.31 /0.6901	27.58 /0.7413	**27.99 /0.7783**	27.80 /0.7591	27.96 /0.7724
σ=50	25.60 /0.6866	26.04 /0.6967	25.15 /0.6330	24.49 /0.5890	24.58 /0.5944	25.08 /0.6215	25.55 /0.6639	25.47 /0.6542	**25.73 /0.6790**

'(LM, BSD)': total GAN trained on LabelMe dataset, and trained denoiser only on BSD set.

with iterative learning, $G2G_I$ could effectively learn with increased dataset capacity.

  - For mixture noise and correlated noise on BSD68

PSNR /SSIM			BM3D	DnCNN-B	N2N	N2V	GCBD	G2G	$G2G_I$
Mixture Noise	Case A	s=15	41.44 /0.9822	38.95 /0.9687	39.81 /0.9737	31.90 /0.9087	42.00 / -	42.55 /0.9881	**42.90 /0.9900**
		s=25	37.97 /0.9647	36.97 /0.9555	37.12 /0.9589	31.05 /0.8909	39.87 / -	39.99 /0.9827	**40.30 /0.9845**
	Case B	s=30	30.12 /0.8496	30.39 /0.8549	30.38 /0.8607	27.85 /0.7679	-	30.37 /0.8456	**30.66 /0.8606**
		s=50	29.27 /0.8190	30.05 /0.8474	30.03 /0.8507	23.48 /0.5157	-	29.93 /0.8330	**30.16 /0.8491**
Correlated Noise		σ=15	29.84 /0.8504	30.71 /0.8916	30.66 /0.8950	27.98 /0.8046	-	30.88 /0.8879	**31.41 /0.8890**
		σ=25	26.69 /0.7544	27.36 /0.8225	27.31 /0.8234	25.76 /0.7237	-	27.38 /0.8007	**27.85 /0.9017**

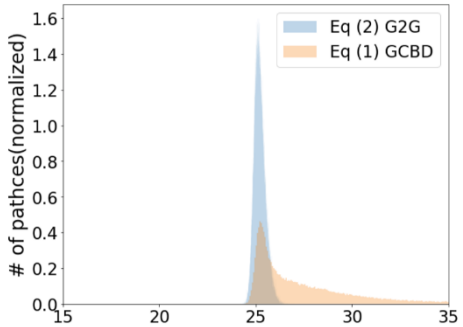
Case A: 70% N(0, 0.01), 20% N(0, 1), 10% Uniform[-s, s]
Case B: 70% N(0, 15), 20% N(0, 25), 10% Uniform[-s, s]

Correlated noise: weighted sum of N(0, σ^2) noise and average of the noise in neighboring patch.
This noise is not spatially-independent.

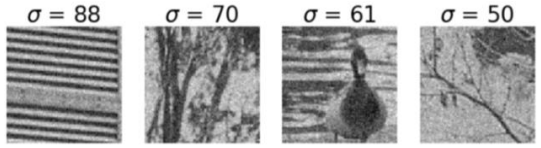
(2019, S. Cha, et al., 'GAN2GAN: Generative Noise Learning for Blind Image Denoising with Single Noisy Images')

GAN2GAN

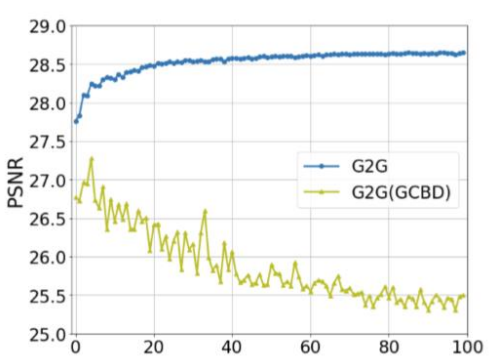
- Results
  - Comparison between its noisy path extraction and that of GCBD



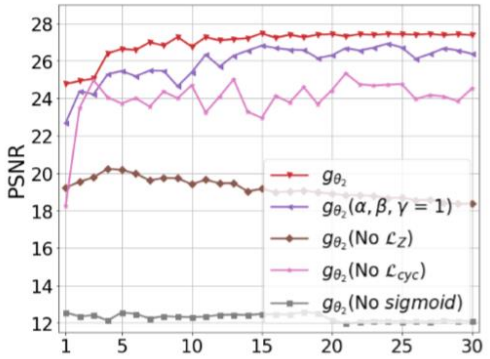
Extracted noise block's empirical variance is mostly correct,
but those of GCBD usually fails because it extracts patches with high frequencies like:



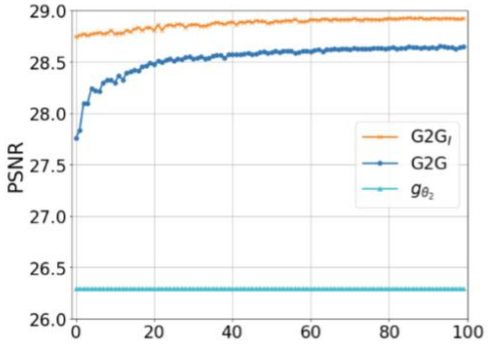
  - Ablation studies



Effect of noisy patch extraction



Effect of different architectures



Effect of iterative GAN2GAN

GRDN

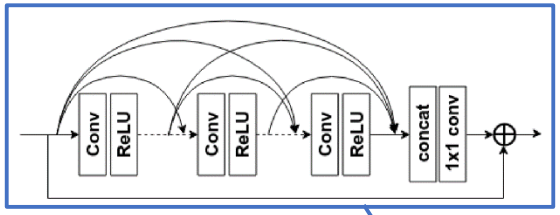
GRDN:Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling

Dong-Wook Kim*, Jae Ryun Chung*, and Seung-Won Jung
Department of Multimedia Engineering
Dongguk University, 04620, Seoul, Korea
spnova12@gmail.com, wjdwofus1004@gmail.com, swjung83@gmail.com

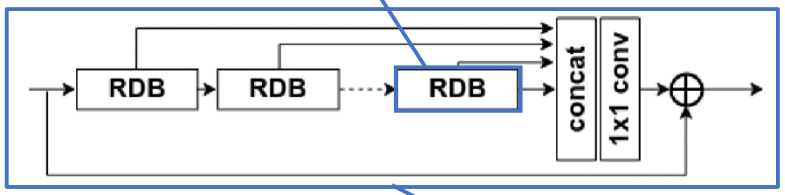
- NTIRE2019 Real Image Denoising Challenge: sRGB track winner
  - Also a GAN-based method,
  - with consideration of signal-dependent noise and camera metadata.
  - Used highly dense-connected architecture.

GRDN

- Model architecture
  - Used dense residual blocks (RDB) and further designed GRDB with it, which is cascaded to form the total architecture.



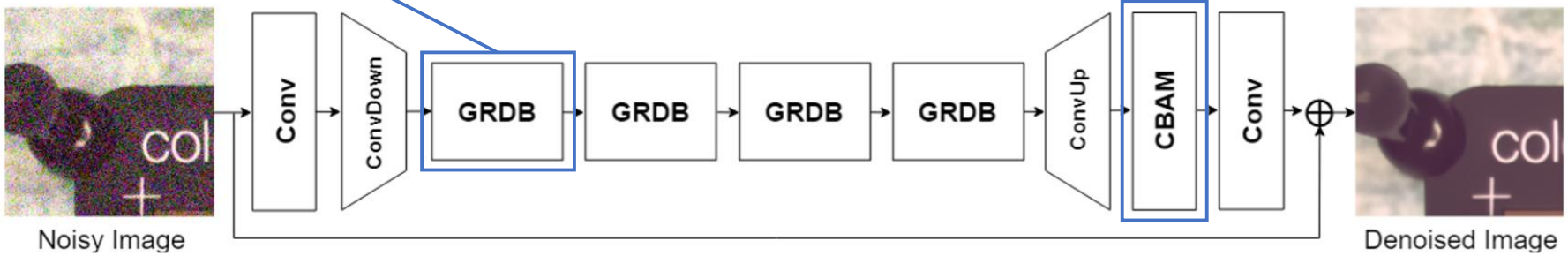
RDB



GRDB

- Adopted attention module, CBAM* which contains sequential channel and spatial attention.

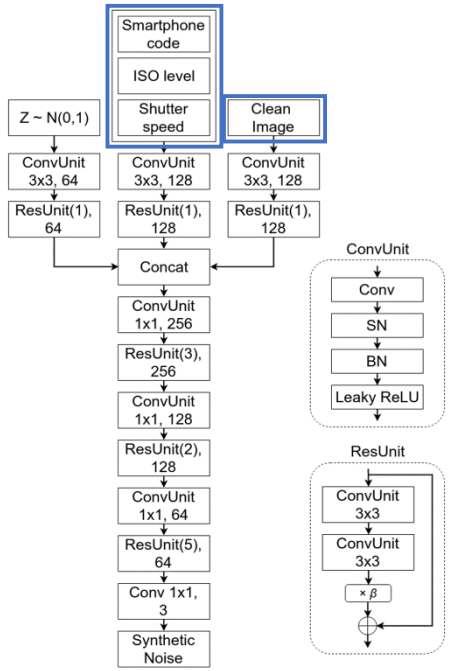
*(2019, S. Woo, et al., 'CBAM: Convolution Block Attention Module')



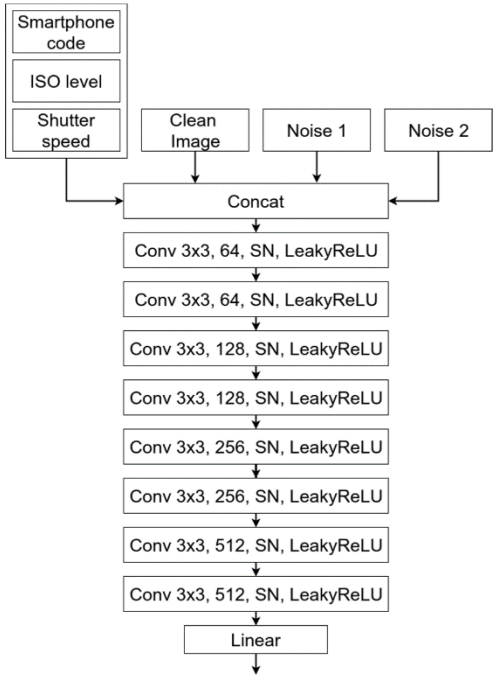
(2019, D. Kim, et al., 'GRDN: Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling')

GRDN

- Training GAN with conditioning signals
  - Its generator also gets camera metadata(it is given in test phase) and clean image as its inputs
  - With this, it can generate signal-dependent noise(like photon noise) map.



Generator

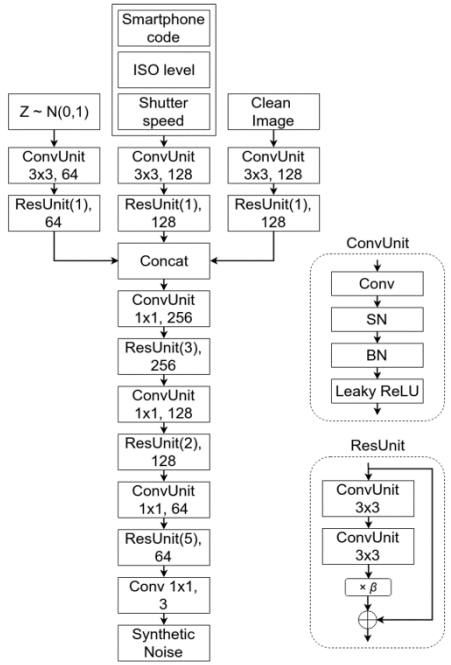


Discriminator

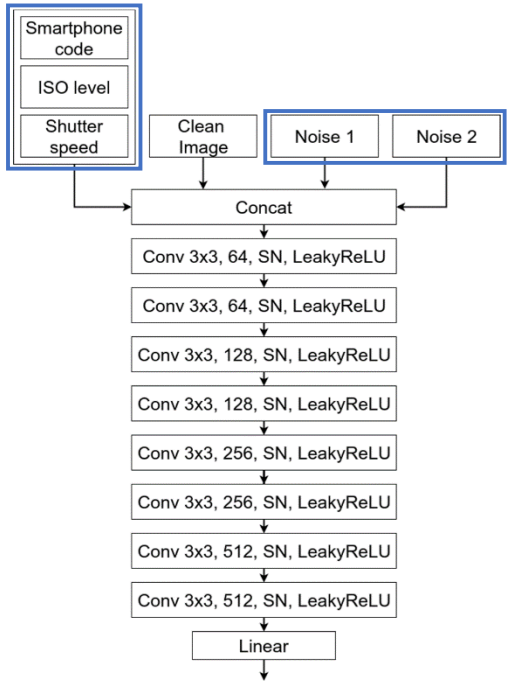
(2019, D. Kim, et al., 'GRDN: Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling')

GRDN

- Training GAN with conditioning signals
  - Its discriminator explicitly compares the generated noise map and corresponding real one.
  - Also takes camera information as its input.



Generator

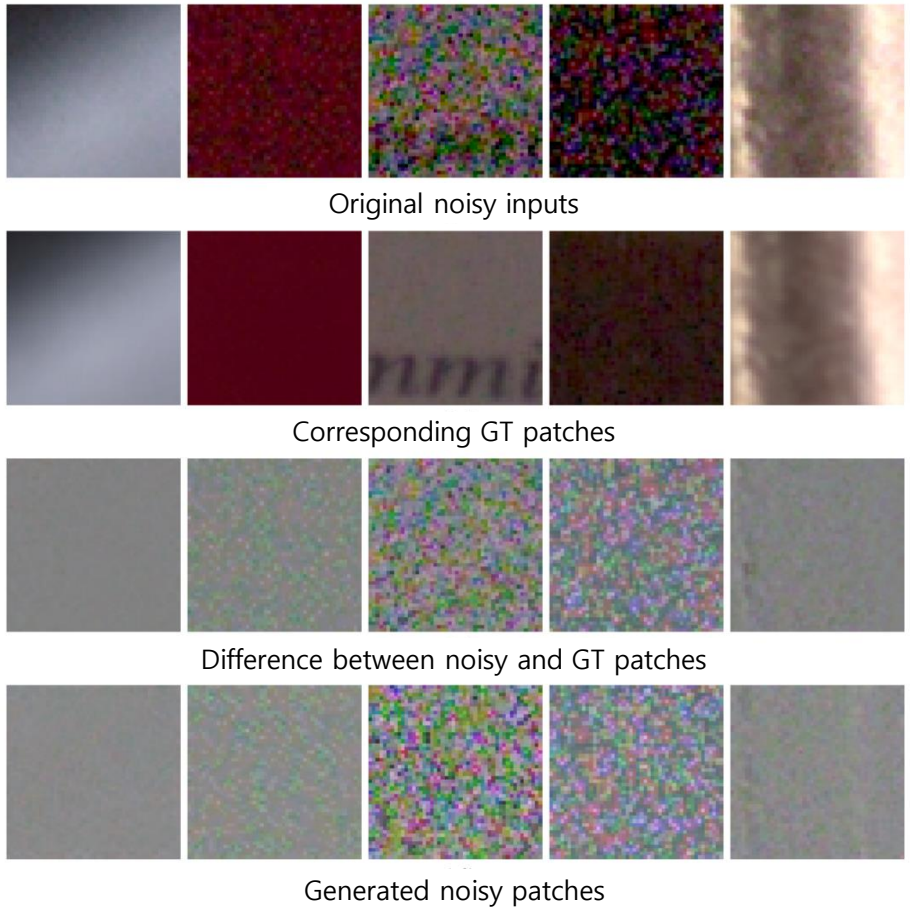


Discriminator

(2019, D. Kim, et al., 'GRDN: Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling')

GRDN

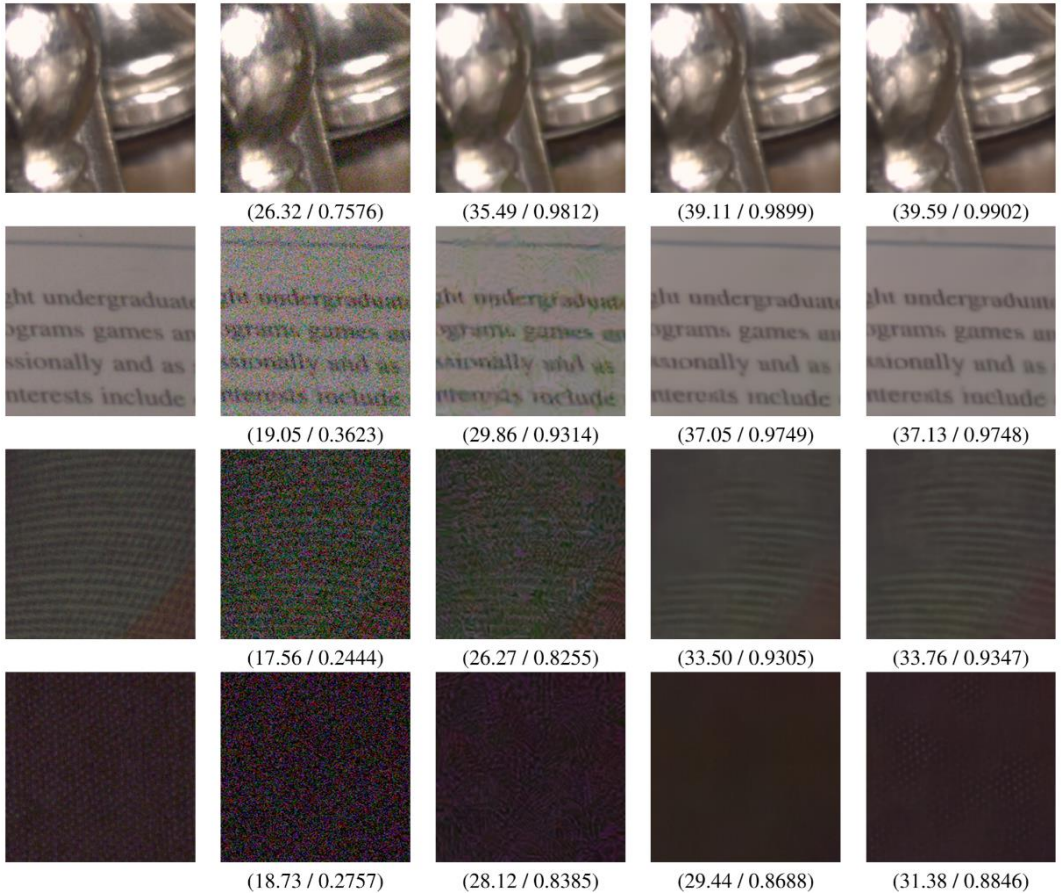
- Generated noise samples



(2019, D. Kim, et al., 'GRDN: Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling')

GRDN

- Results on SIDD dataset



Method	PSNR	SSIM
GRDN (Ours)	**39.931743**	**0.973589**
2nd method	39.883139	0.973113
3rd method	39.818198	0.972963
4th method	39.675235	0.972554
5th method	39.610533	0.972637

Performance comparison
for NTIRE2019 Real Image Denoising Challenge: sRGB track

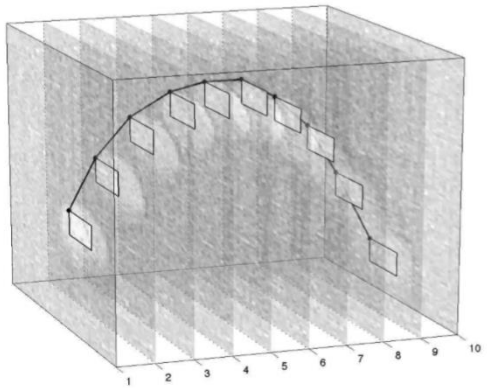
*(2018, Y. Zhang, et al., 'Residual Dense Network for Image Super-Resolution')

(2019, D. Kim, et al., 'GRDN: Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling')

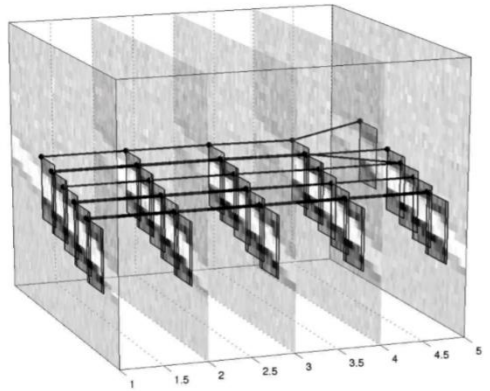
Video Denoising

- Now introduce some important works in deep **video denoising**.

- One of traditional methods: V-BM4D



A volume got from a trajectory



A group of mutually similar volumes

- Inherited from BM3D.

1. Find a spatiotemporal volume along time,
2. Form a group by stacking those volumes along a fourth dimension.
3. Apply 4-D collaborative filtering.

+) 'V-BM3D': volumes are gathered without distinguishing temporal and spatial similarity. (without fourth dimension)

(2012, M. Maggioni, et al., 'Video Denoising, Deblocking, and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms')

VNLnet

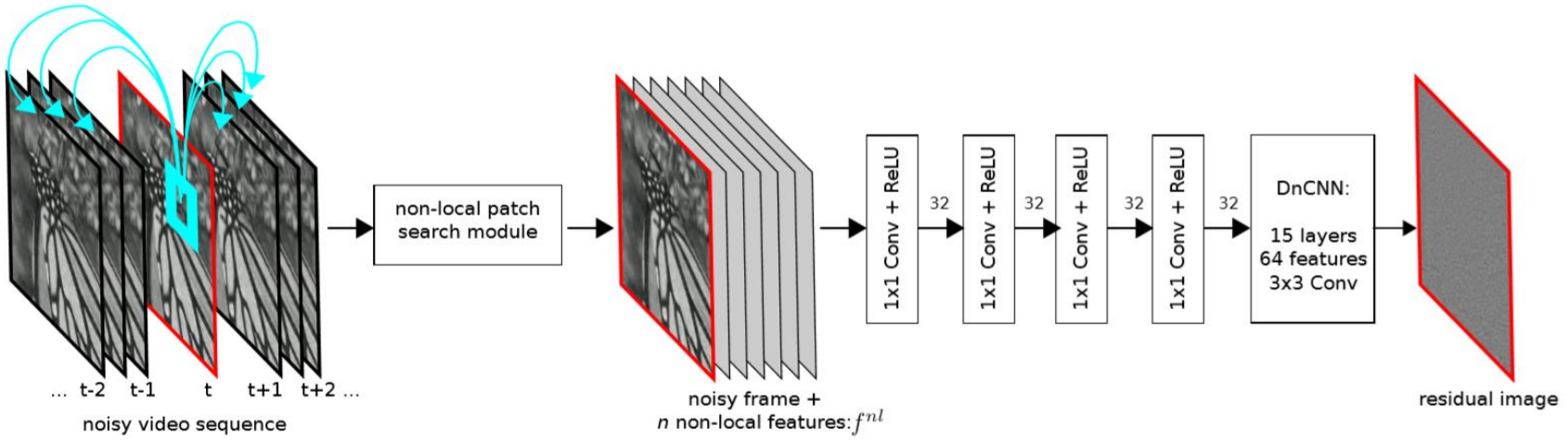
Non-Local Video Denoising by CNN

Axel Davy Thibaud Ehret Gabriele Facciolo
Jean-Michel Morel Pablo Arias
CMLA, ENS Cachan, CNRS
Université Paris-Saclay, 94235 Cachan, France
axel.davy@ens-cachan.fr

- Using temporal self-similarity for deep video denoising
  - Finds similar patches from neighboring frames.
  - Feed-forward non-local feature vector to CNN.

VNLnet

- For patch $v_{x,t}$ centered at (pixel x, time t), search similar patches in 3D window $w_{row} \times w_{col} \times w_t$.
- The gathered non-local feature vector, $f^{\mathrm{nl}}(x,t) = [v(x_1,t_1), ..., v(x_n,t_n)]$ with n channels, is the input to the CNN(few layers added to DnCNN).

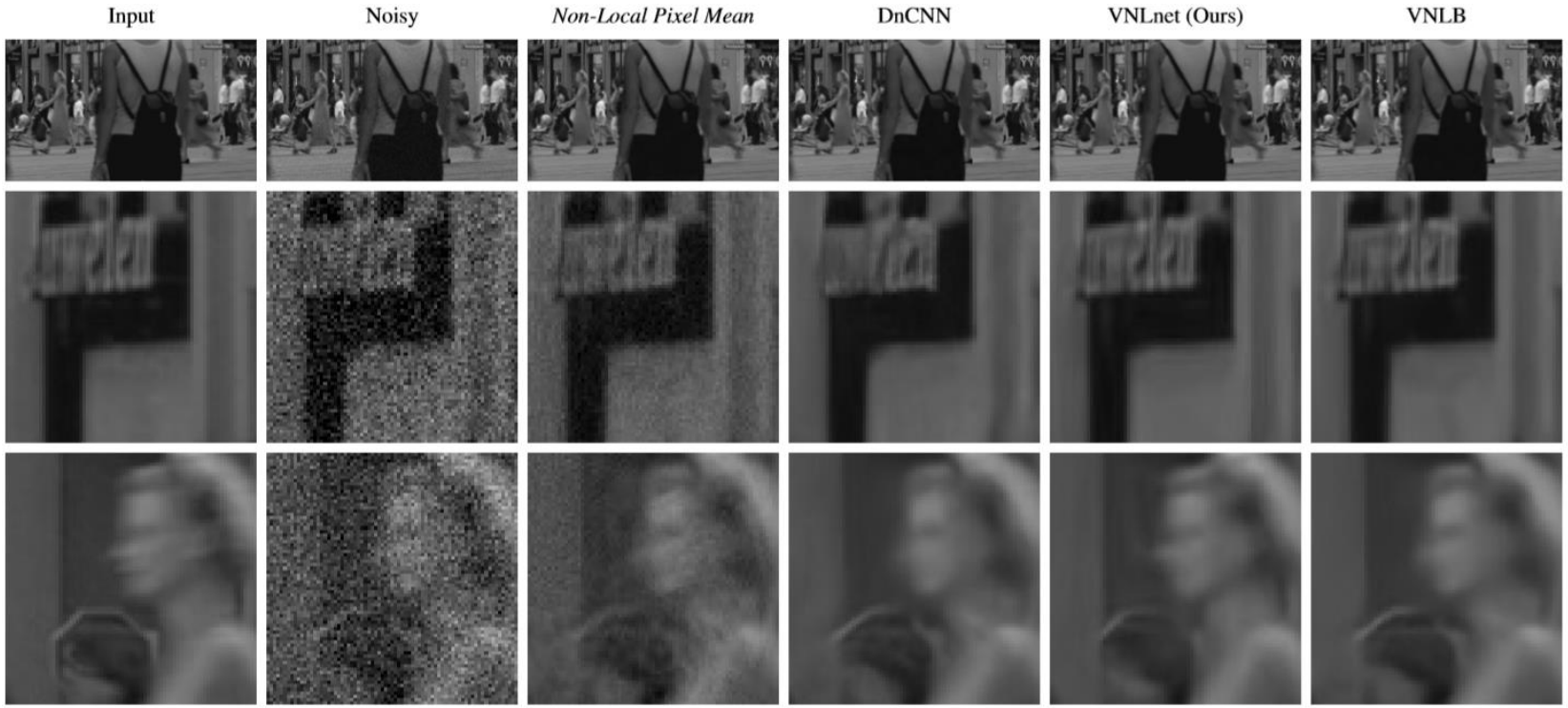


- Here they implemented the non-local patch search module with n-nearest neighbor algorithm.

(2018, A. Davy, et al., 'Non-Local Video Denoising by CNN')

VNLnet

- Results
  - Trained on scaled 960*540 size videos from YouTube
  - Tested on Derf's Test Media(https://media.xiph.org/video/derf/) collection and DAVIS 2017 video segmentation challenge dataset(https://davischallenge.org/challenge2017/index.html)



Result comparison on noise std 20

(2018, A. Davy, et al., 'Non-Local Video Denoising by CNN')

VNLnet

- Its performance is comparable with VNLB*, the best performing method.

σ	Method	crowd	park joy	pedestrians	station	sunflower	touchdown	tractor	average
10	SPTWO	36.57 / .9651	35.87 / .9570	41.02 / .9725	41.24 / .9697	42.84 / .9824	40.45 / .9557	**38.92** / .9701	39.56 / .9675
	VBM3D	35.76 / .9589	35.00 / .9469	40.90 / .9674	39.14 / .9651	40.13 / .9770	39.25 / .9466	37.51 / .9575	38.24 / .9599
	VBM4D	36.05 / .9535	35.31 / .9354	40.61 / .9712	40.85 / .9466	41.88 / .9696	39.79 / .9440	37.73 / .9533	38.88 / .9534
	VNLB	**37.24** / **.9702**	**36.48** / **.9622**	**42.23** / **.9782**	**42.14** / **.9771**	**43.70** / **.9850**	**41.23** / **.9615**	**40.20** / **.9773**	**40.57** / **.9731**
	DnCNN	34.39 / .9455	33.82 / .9329	39.46 / .9641	37.89 / .9412	40.20 / .9702	38.28 / .9269	36.91 / .9568	37.28 / .9482
	VNLnet	**36.90** / **.9711**	**36.19** / **.9642**	**41.66** / **.9760**	**41.94** / **.9735**	**43.58** / **.9849**	**40.75** / **.9575**	38.78 / **.9709**	**39.97** / **.9712**
20	SPTWO	32.94 / .9319	32.35 / .9161	37.01 / .9391	**38.09** / **.9461**	38.83 / .9593	**37.55** / **.9287**	**35.15** / .9363	35.99 / .9368
	VBM3D	32.34 / .9093	31.50 / .8731	37.06 / .9423	35.91 / .9007	36.25 / .9393	36.17 / .9065	33.53 / .8991	34.68 / .9100
	VBM4D	32.40 / .9126	31.60 / .8832	36.72 / .9344	36.84 / .9224	37.78 / .9517	36.44 / .9034	33.95 / .9104	35.10 / .9169
	VNLB	**33.49** / **.9335**	**32.80** / **.9154**	**38.61** / **.9583**	**38.78** / **.9470**	**39.82** / **.9698**	**37.47** / **.9220**	**36.67** / **.9536**	**36.81** / **.9428**
	DnCNN	30.47 / .8890	30.03 / .8625	35.81 / .9302	34.37 / .8832	36.19 / .9361	35.35 / .8782	32.99 / .9019	33.60 / .8973
	VNLnet	**33.19** / **.9367**	**32.57** / **.9207**	**37.96** / **.9528**	37.87 / .9379	**39.53** / **.9667**	36.79 / .9040	35.06 / **.9367**	**36.14** / **.9365**
40	SPTWO	29.02 / .8095	28.79 / .8022	31.32 / .7705	32.37 / .7922	32.61 / .7974	31.80 / .7364	30.61 / .8223	30.93 / .7901
	VBM3D	28.73 / .8295	27.93 / .7663	33.00 / .8828	32.57 / .8239	32.39 / .8831	33.38 / **.8624**	29.80 / .8039	31.11 / .8360
	VBM4D	28.72 / .8339	27.99 / .7751	32.62 / .8683	32.93 / .8441	33.66 / .8999	33.68 / .8603	30.20 / .8205	31.40 / .8432
	VNLB	**29.88** / **.8682**	**29.28** / **.8309**	**34.68** / **.9167**	**34.65** / **.8871**	**35.44** / **.9329**	**34.18** / **.8712**	**32.58** / **.8921**	**32.95** / **.8856**
	DnCNN	26.85 / .7979	26.65 / .7525	32.01 / .8660	30.96 / .7899	32.13 / .8705	32.78 / .8346	29.25 / .7976	30.09 / .8156
	VNLnet	**29.46** / **.8650**	**28.96** / **.8289**	**33.88** / **.9027**	**33.48** / **.8577**	**34.65** / **.9150**	**33.70** / .8465	**31.17** / **.8596**	**32.19** / **.8679**

Results on Derf's Test Media collection

Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 40$
DnCNN	36.80	32.94	28.69
VBM3D	37.43	33.75	30.12
VNLnet	**39.08**	**35.44**	**31.79**

Results on DAVIS 2017 dataset

- Using CNN, it is faster.
  - Though it doesn't directly compares its running time on CPU-powered environment because it used GPU-powered implementation of patch searching.

VBM3D	DnCNN	VBM4D	VNLB
1.3s	13s	52s	140s

Running time per frame(960*540) on CPU

Non-local search	Rest of the network	DnCNN
932 ms	80 ms	95 ms

Running time of proposed method on GPU, compared with DnCNN

(2018, A. Davy, et al., 'Non-Local Video Denoising by CNN')

Frame-to-Frame Training

Model-blind Video Denoising Via Frame-to-frame Training

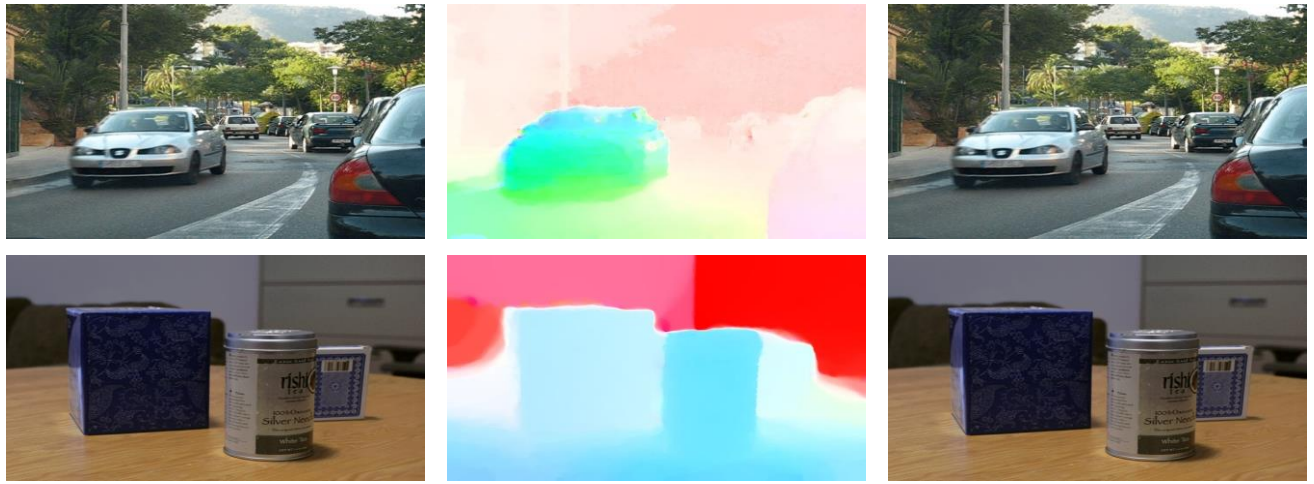
Thibaud Ehret Axel Davy Jean-Michel Morel
Gabriele Facciolo Pablo Arias
CMLA, ENS Cachan, CNRS
Université Paris-Saclay, 94235 Cachan, France
thibaud.ehret@ens-cachan.fr

- Model-blind video denoising
  - With temporal redundancy in videos, applied Noise2Noise learning.
  - Insists it can perform well with online learning

(2018, T. Ehret, et al., 'Model-blind Video Denoising Via Frame-to-frame Training')

Frame-to-Frame Training

- Frame-to-Frame Noise2Noise learning
  - Need independent noisy observation pair from same underlying scene.
  - Consider consecutive frames in video **as same clean signal transformed by motion**.
  - Using TV-L1 optical flow*, warp frame f_{t-1} into f_t with optical flow v_t



Input frames (animated)

Optical flow

Warped frame and actual frame (animated)

(visualization by C. Liu, et al., from 'Human-Assisted Motion Annotation')

  - To exclude occluded parts from loss calculation, masked out pixels with high divergence in optical flow

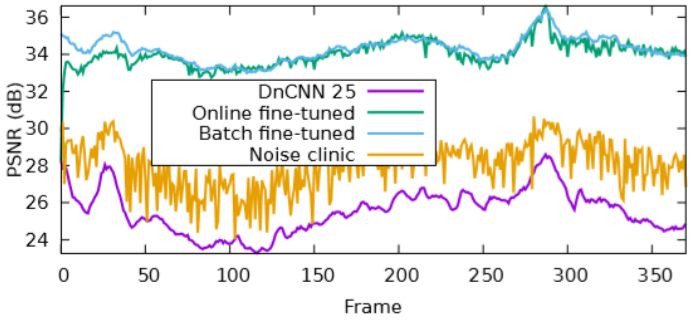
$$\kappa_t(x) = \begin{cases} 0 \text{ if } |\text{div } v_t(x)| > \tau \\ 1 \text{ if } |\text{div } v_t(x)| \leq \tau. \end{cases}$$

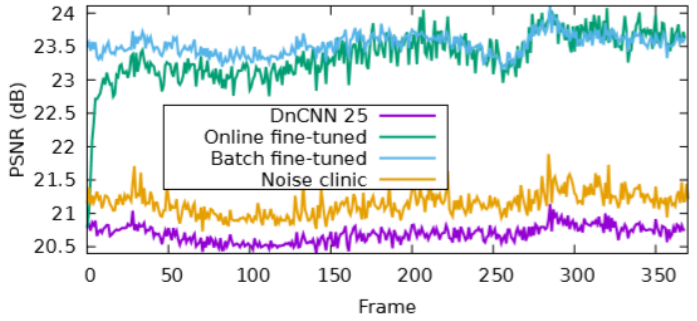
(2018, T. Ehret, et al., 'Model-blind Video Denoising Via Frame-to-frame Training')

Frame-to-Frame Training

- Better results for various noise types
  - It assumes a situation **without large video dataset**, starting **from pretrained DnCNN**.
  - '**Batch fine-tuned**': for a test video, it fine-tunes with samples over all frames in it.
  - '**Online fine-tuned**': frames are given sequentially, train the network for current pair (f_t, f_{t-1}^{warped}) with $N = 20$ steps.

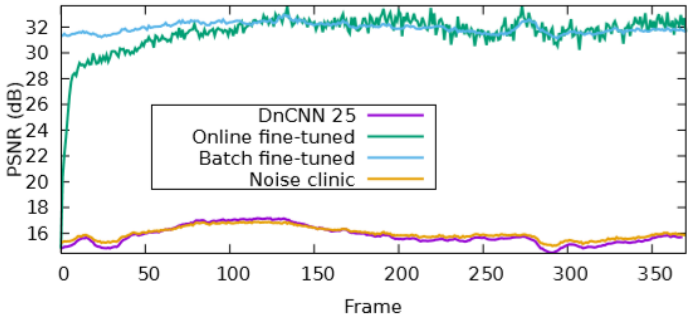
for multiplicative Gaussian

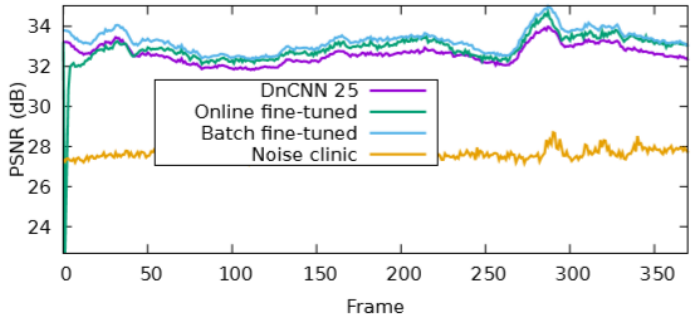




for spatially-correlated Gaussian

for salt and pepper





for Gaussian with JPEG compression

Noise Clinic: (2015, M. Lebrun, et al., The Noise Clinic: a Blind Image Denoising Algorithm)
(2018, T. Ehret, et al., 'Model-blind Video Denoising Via Frame-to-frame Training')

Frame-to-Frame Training

- Result comparison (example) for Gaussian with JPEG compression

(2018, T. Ehret, et al., 'Model-blind Video Denoising Via Frame-to-frame Training')

Frame-to-Frame Training

- It could handle many types of noise without severe artifacts.
  - Though the right loss term should be chosen, just like Noise2Noise.

(2018, T. Ehret, et al., 'Model-blind Video Denoising Via Frame-to-frame Training')

ViDeNN

ViDeNN: Deep Blind Video Denoising

Michele Claus
Computer Vision lab
Delft University of Technology
claus.michele@hotmail.it

Jan van Gemert
Computer Vision lab
Delft University of Technology
http://jvgemert.github.io/

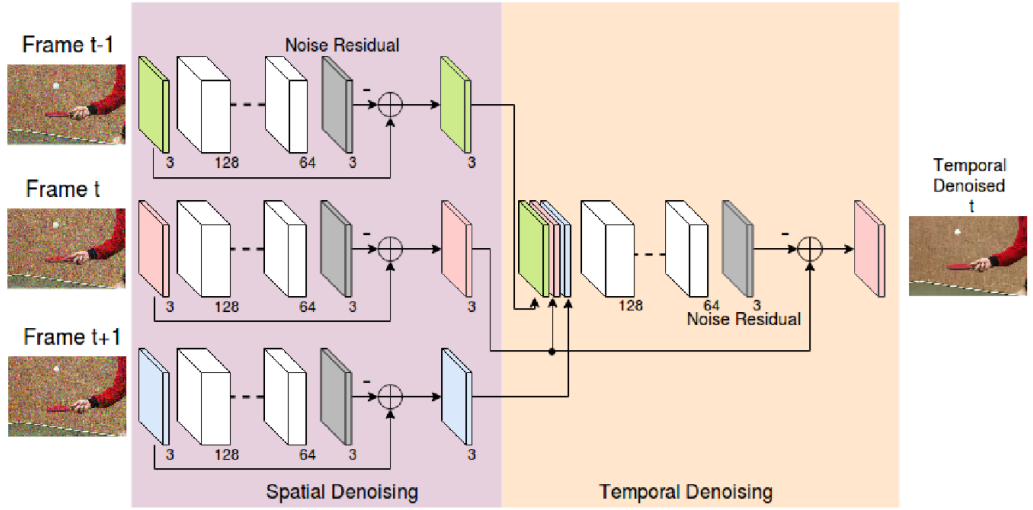
- Supervised learning for deep video denoising, with higher performance*.
  - Single frame denoising followed by temporal denoising
  - Targets realistic noise.

*(2016, X. Chen, et al., 'Deep RNNs for Video Denoising') showed performance lower than traditional VBM3D or VBM4D.

(2019, M. Claus, et al., 'ViDeNN: Deep Blind Video Denoising')

ViDeNN

- Total model
  - Two parts: Single frame denoising(Spatial CNN), temporal denoising(Temp3-CNN)
  - **Spatial CNN** is trained as a single image denoiser, with Waterloo dataset.
  - **Temp3-CNN** is trained with Derf's collection, synthetically adding noise.



  - This architecture is designed with extensive ablation studies introduced.

(2019, M. Claus, et al., 'ViDeNN: Deep Blind Video Denoising')

ViDeNN

- Targeting realistic noise
  - With basic AWGN, also used synthesized analog-to-digital converter noise:

$$M = \sqrt{\underbrace{\frac{Ag * Dg}{N_{sat} * s}}_{\text{PSN}} + \underbrace{Dg^2 * (Ag * CT1_n + CT2_n)^2}_{\text{Read Noise}}}$$

$$\text{Noisy Image} = s + \mathcal{N}(0,1) * M(s),$$

    - It's signal-dependent, controlled by random A_g(analog gain) and D_g(digital gain) and the constants(N_{sat}, $CT1_n$, $CT2_n$)
    - Used for training Temp3-CNN

  - For low-light data, with Renoir set, made their own dataset
    - Simply took many raw images and averaged them. (without other processing)

(2019, M. Claus, et al., 'ViDeNN: Deep Blind Video Denoising')

ViDeNN

- Results on Derf's collection
  - 'ViDeNN-G' means it is specifically trained only on Gaussian noise, unlike ViDeNN.

	Tennis			*Old Town Cross*			*Park Run*			*Stefan*		
Res./Frames	240×352 / 150			720×1280 / 500			720×1280 / 504			656×1164 / 300		
σ	5	25	40	15	25	40	15	25	40	15	25	55
ViDeNN	35.51	29.97	28.00	32.15	30.91	29.41	31.04	28.44	25.97	32.06	29.23	24.63
ViDeNN-G	**37.81**	**30.36**	**28.44**	32.39	**31.29**	**29.97**	**31.25**	**28.72**	**26.36**	**32.37**	**29.59**	**25.06**
VBM4D [10]	34.64	29.72	27.49	**32.40**	31.21	29.57	29.99	27.90	25.84	29.90	27.87	23.83
CBM3D [35]	27.04	26.37	25.62	28.19	27.95	27.35	24.75	24.46	23.71	26.19	25.89	24.18
DnCNN [14]	35.49	27.47	25.43	31.47	30.10	28.35	30.66	27.87	25.20	32.20	29.29	24.51

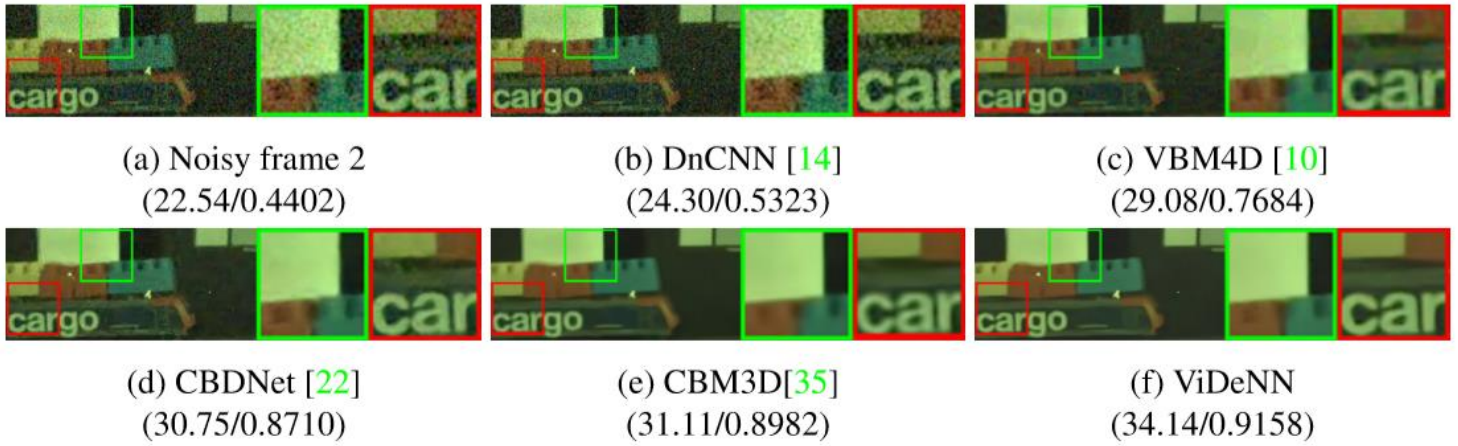
(a) Noisy frame 148 (b) Noisy frame 149 (c) Noisy frame 150 (d) CBM3D[6] (25.60/0.9482) (e) VBM4D[10] (24.51/0.9292) (f) ViDeNN-G (27.19/0.9605)

(2019, M. Claus, et al., 'ViDeNN: Deep Blind Video Denoising')

ViDeNN

- Results on low-light video set
  - Low-light video dataset: manually recorded sequences with specific camera.
  - From three scenes, two light intensity levels each.

	Train		Mountains	Windmill		
Res./Frames	212×1091 / 4		1080×1920 / 4	1080×1920 / 3		
Light	50/255	55/255	[55,75]/255	44.6 lux	118 lux	212 lux
ViDeNN	**34.05**	**36.96**	**40.84**	**32.96**	**35.42**	**36.69**
VBM4D[10]	29.10	33.48	37.34	26.62	30.69	32.92
CBDNet[22]	30.89	34.56	39.91	29.56	34.31	36.22
CBM3D[35]	31.27	34.06	40.20	29.81	34.06	35.74
DnCNN[14]	24.33	29.87	32.39	21.73	25.55	27.87



(a) Noisy frame 2 (22.54/0.4402) (b) DnCNN [14] (24.30/0.5323) (c) VBM4D [10] (29.08/0.7684)

(d) CBDNet [22] (30.75/0.8710) (e) CBM3D[35] (31.11/0.8982) (f) ViDeNN (34.14/0.9158)

(2019, M. Claus, et al., 'ViDeNN: Deep Blind Video Denoising')

Conclusion: Issues In Short

- Deep denoising is evolving in many aspects:
  - Powerful, efficient architecture design
  - Handling real noise
    - Getting dataset
    - Generative models to mimic it
  - Unsupervised learning
    - Based on statistical characteristic
    - Applying traditional denoising idea to deep learning
  - Total image restoration
    - with super-resolution, deblurring, illumination, etc.
  - Many types of target data
    - Video, fast burst images, medical images, hyperspectral images, etc.

https://cv.snu.ac.kr/