(CVPR2019)

Noise2Void - Learning Denoising from Single Noisy Images

Alexander Krull[1,2], Tim-Oliver Buchholz[2], Florian Jug

[1] krull@mpi-cbg.de

[2] Authors contributed equally

MPI-CBG/PKS (CSBD), Dresden, Germany

Introduction

- Task

  Learning image denoising networks from single noisy image.

- Limitation of existing methods

  A. Supervised learning

     Require large amount of paired noisy-clean images

  B. Noise2noise

     - ➢ N2N training requires the availability of pairs of noisy images
     - ➢ the acquisition of such pairs with (quasi) constant ***s*** is only possible for (quasi) static scenes

Method

- Formulation

Given: $x = s + n$

$$p(\boldsymbol{s}, \boldsymbol{n}) = p(\boldsymbol{s})p(\boldsymbol{n}|\boldsymbol{s})$$

$$p(\boldsymbol{s}_i|\boldsymbol{s}_j) \neq p(\boldsymbol{s}_i)$$

$$p(\boldsymbol{n}|\boldsymbol{s}) = \prod_i p(\boldsymbol{n}_i|\boldsymbol{s}_i)$$

$$\mathbb{E}\left[\boldsymbol{n}_i\right] = 0 \quad \mathbb{E}\left[\boldsymbol{x}_i\right] = \boldsymbol{s}_i$$

- Supervised training

$$f(\boldsymbol{x}_{\mathrm{RF}(i)}; \boldsymbol{\theta}) = \hat{\boldsymbol{s}}_i$$

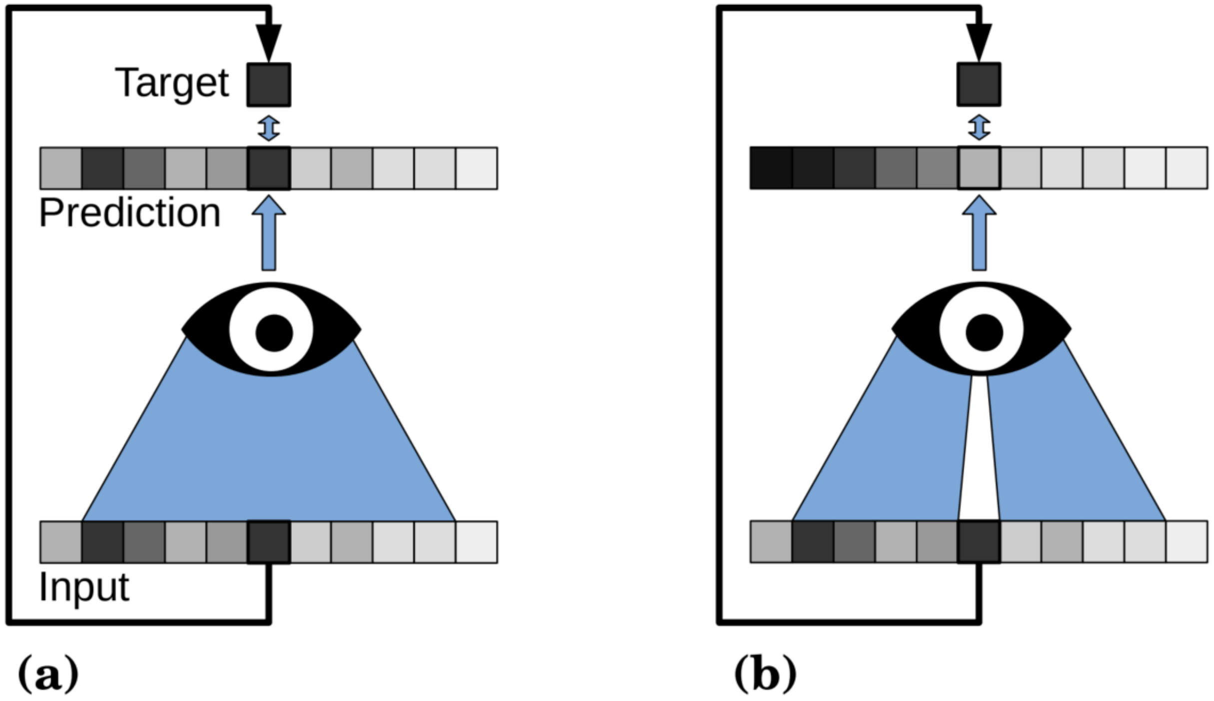
$$\arg\min_{\boldsymbol{\theta}} \sum_j \sum_i L\left(f(\boldsymbol{x}^j_{\mathrm{RF}(i)}; \boldsymbol{\theta}) = \hat{\boldsymbol{s}}^j_i, \boldsymbol{s}^j_i\right)$$

- Noise2noise training

$$\boldsymbol{x}^j = \boldsymbol{s}^j + \boldsymbol{n}^j \text{ and } \boldsymbol{x}'^j = \boldsymbol{s}^j + \boldsymbol{n}'^j$$

Method

- Noise2Void – Blind-spot network



Method

- Noise2Void
  - Replace the value in the center of each input patch with a randomly selected value form the surrounding area

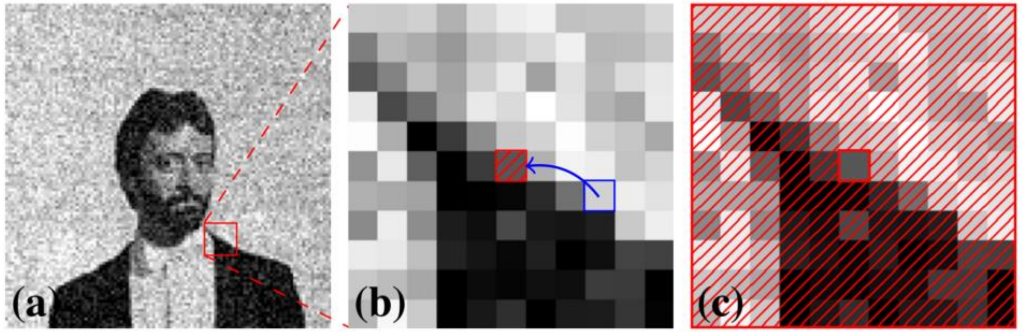
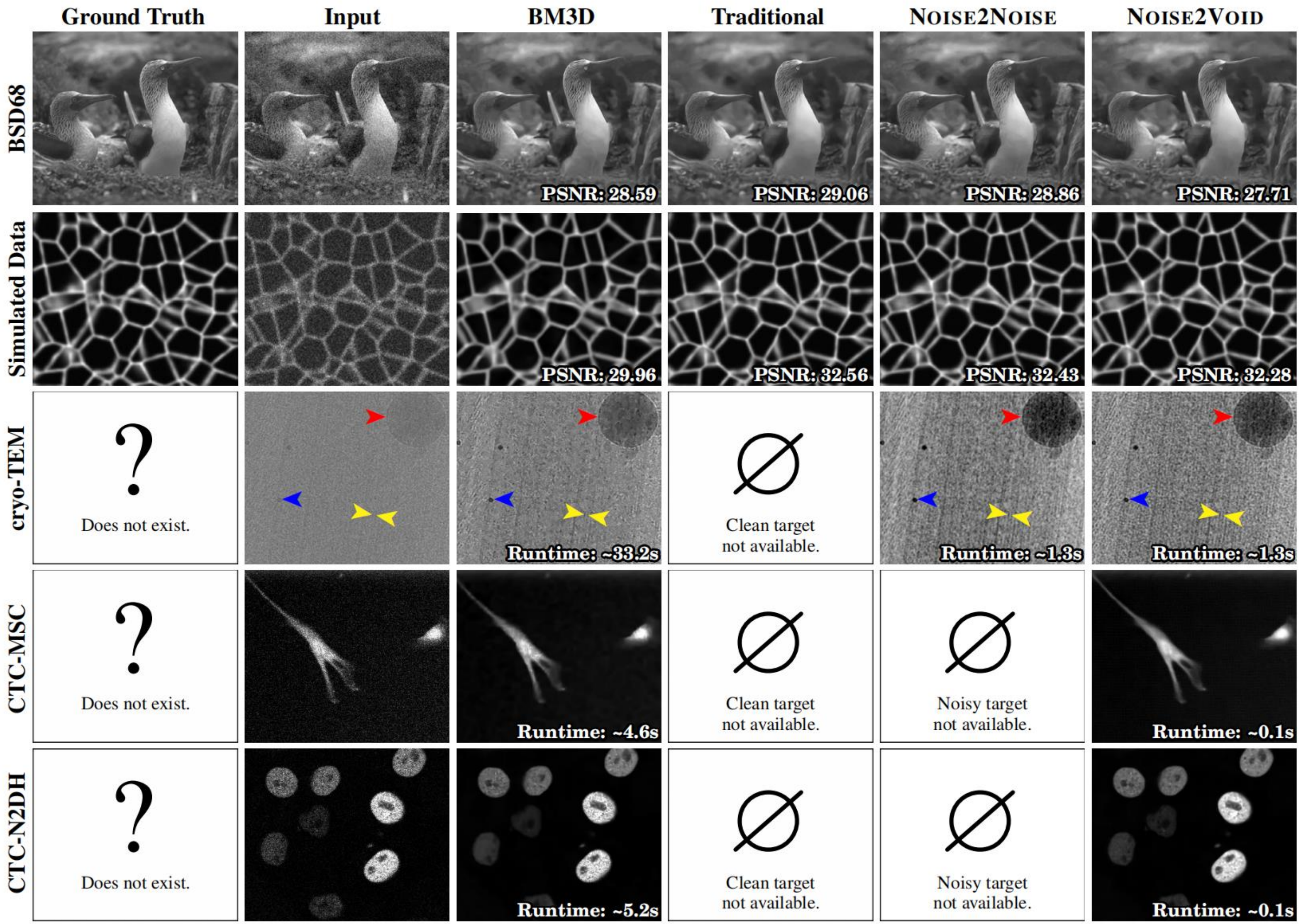


Figure 3: Blind-spot masking scheme used during NOISE2VOID training. **(a)** A noisy training image. **(b)** A magnified image patch from (a). During N2V training, a randomly selected pixel is chosen (blue rectangle) and its intensity copied over to create a blind-spot (red and striped square). This modified image is then used as input image during training. **(c)** The target patch corresponding to (b). We use the original input with unmodified values also as target. The loss is only calculated for the blind-spot pixels we masked in (b).

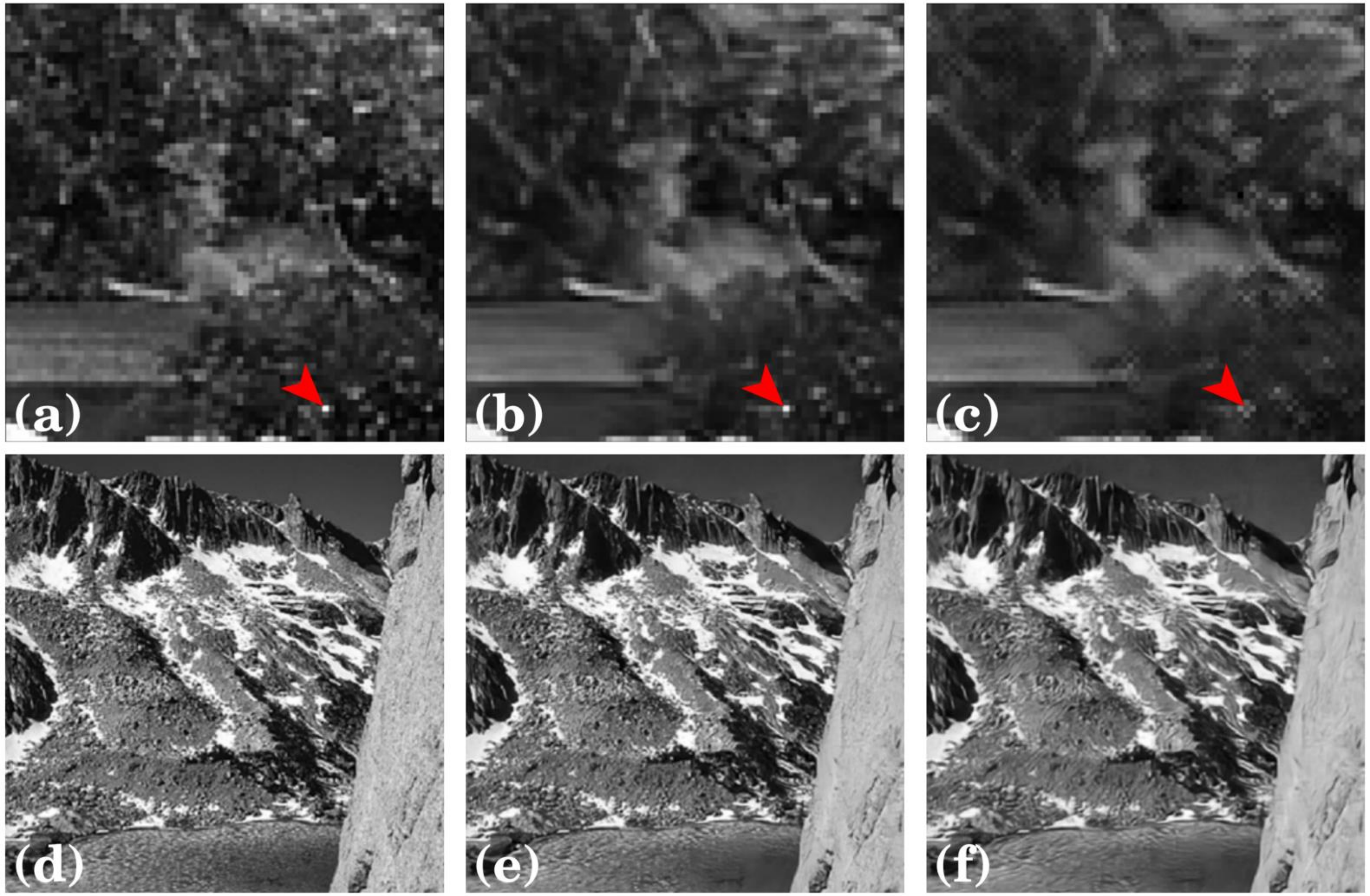
Results

- Comparison with other methods



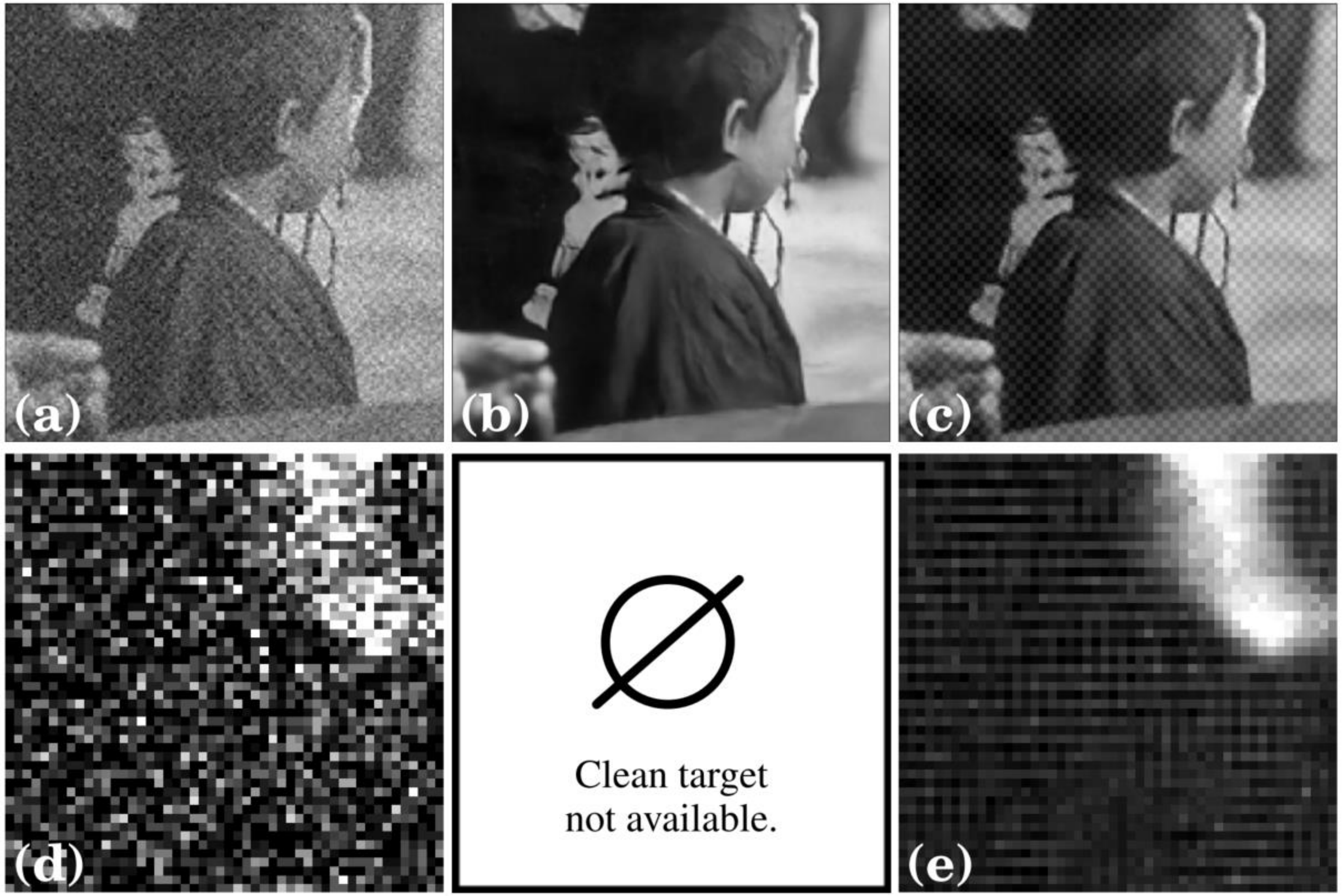
Results

- Limitations



Results

- Limitations



(ECCV2020)

Unpaired Learning of Deep Image Denoising

Xiaohe Wu[1], Ming Liu[1], Yue Cao[1], Dongwei Ren[2], and Wangmeng Zuo[1,3] (✉)

[1] Harbin Institute of Technology, China
[2] University of Tianjin, China
[3] Peng Cheng Lab, China
csxhwu@gmail.com, csmliu@outlook.com,
{cscaoyue,rendongweihit}@gmail.com, wmzuo@hit.edu.com

Introduction

- Task

  Learning blind image denoising networks from an unpaired set of clean and noisy images

- Limitation of existing methods

  A. Supervised learning: require large amount of paired noisy-clean images

  - Synthesize noisy images from noise-less clean images to constitute a paired training set ----- real noise usually is complex, it is difficult to be fully characterized by basic parametric noise model.
  - Designing suitable approaches to acquire the nearly noise-free image ----- the nearly noise-free images may suffer from over-smoothing issue and are cost-expensive to acquire. And such nearly noise-free image acquisition may not be applicable to other imaging mechanisms (e.g., microscopy or medical imaging)

  B. Only use noisy images

  - N2N: requires that the underlying clean images in each pair are exactly the same and the noises are independently drawn from the same distribution;
  - N2V: computationally very inefficient in training, fails to exploit the pixel value at blind spot.
  - Fail to exploit clean images in training

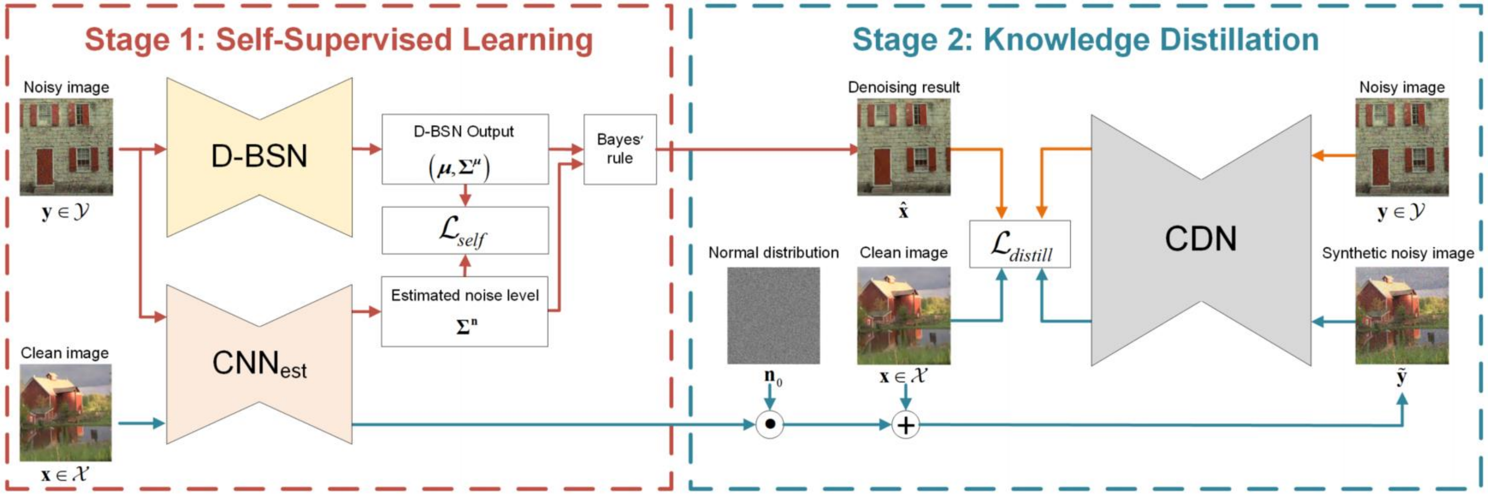
Method

- Two-Stage Training and Knowledge Distillation

Given: **x** denotes a clean image, and **y** is a noisy image, $\mathbf{y} = \tilde{\mathbf{x}} + \mathbf{n}$

Assume: **x** is spatially correlated, **n** is pixel-independent and signal-dependent Gaussian, that is, the noise variance (or noise level) at pixel i is determined only by the underlying noise-free pixel value $\tilde{x}_i$ at pixel i.

$$\mathrm{var}(n_i) = g_{\tilde{\mathbf{x}}}(\tilde{x}_i)$$



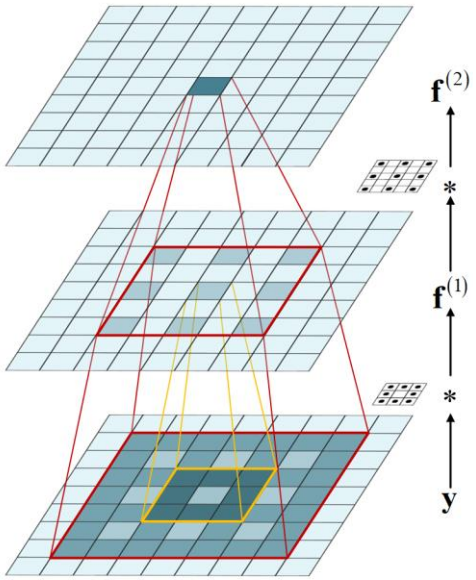
First stage: novel blind-spot network D-BSN + image specific noise model CNN_{est}. $y \rightarrow \hat{x}_y, NLF\ g_y(y)$

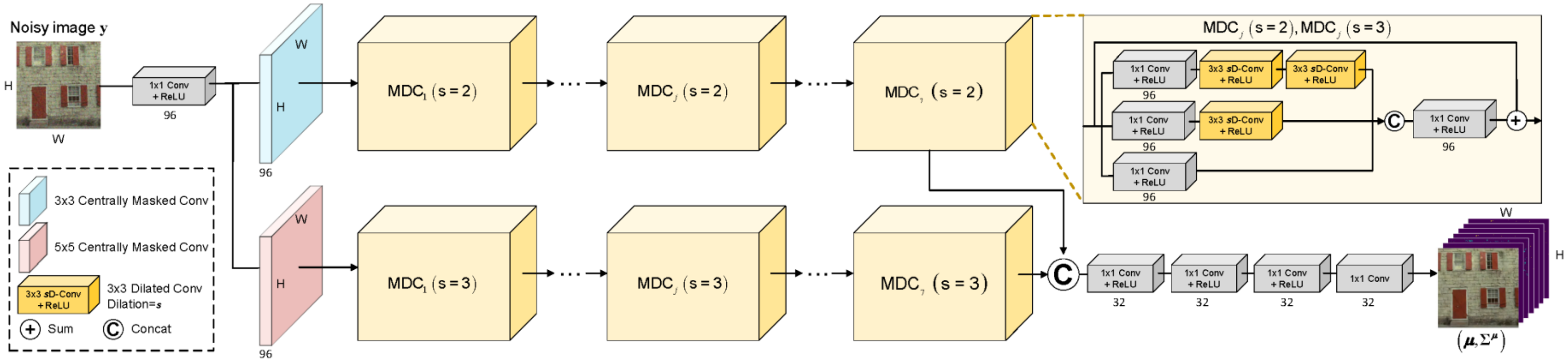
Second stage: two sets of paired noisy-clean images, $\tilde{\mathbf{y}} = \mathbf{x} + g_{\mathbf{y}}(\mathbf{x}) \cdot \mathbf{n}_0$ $\{(\mathbf{x}, \tilde{\mathbf{y}}) | \mathbf{x} \in \mathcal{X}\}\{(\hat{\mathbf{x}}_{\mathbf{y}}, \mathbf{y}) | \mathbf{y} \in \mathcal{Y}\}$

$$\mathcal{L}_{distill} = \sum\nolimits_{\mathbf{x} \in \mathcal{X}} \|\mathrm{CDN}(\tilde{\mathbf{y}}) - \mathbf{x}\|^2 + \lambda \sum\nolimits_{\mathbf{y} \in \mathcal{Y}} \|\mathrm{CDN}(\mathbf{y}) - \hat{\mathbf{x}}_{\mathbf{y}}\|^2$$

Method

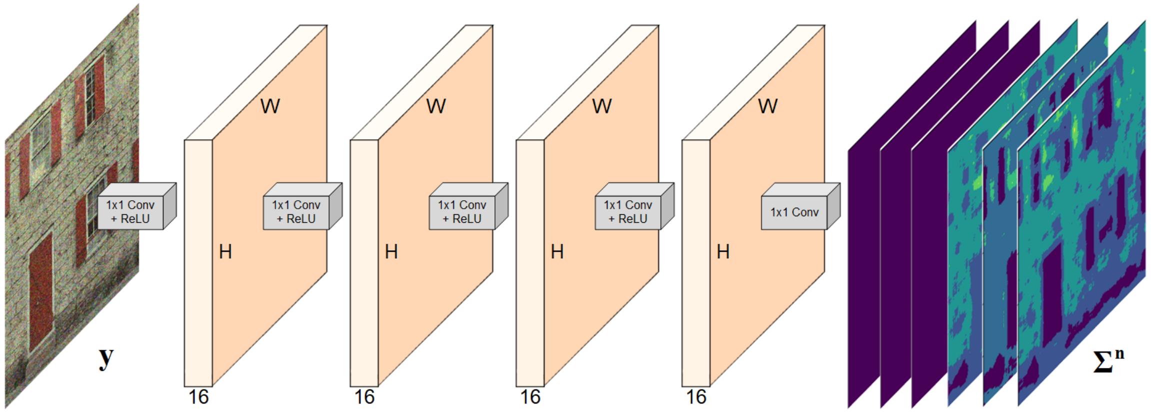
- D-BSN
  - Blind-spot requirement: exclude the effect of the input value at the same position.
  - D-BSN: centrally masked convolution, dilated convolution, and 1*1 convolution





Method

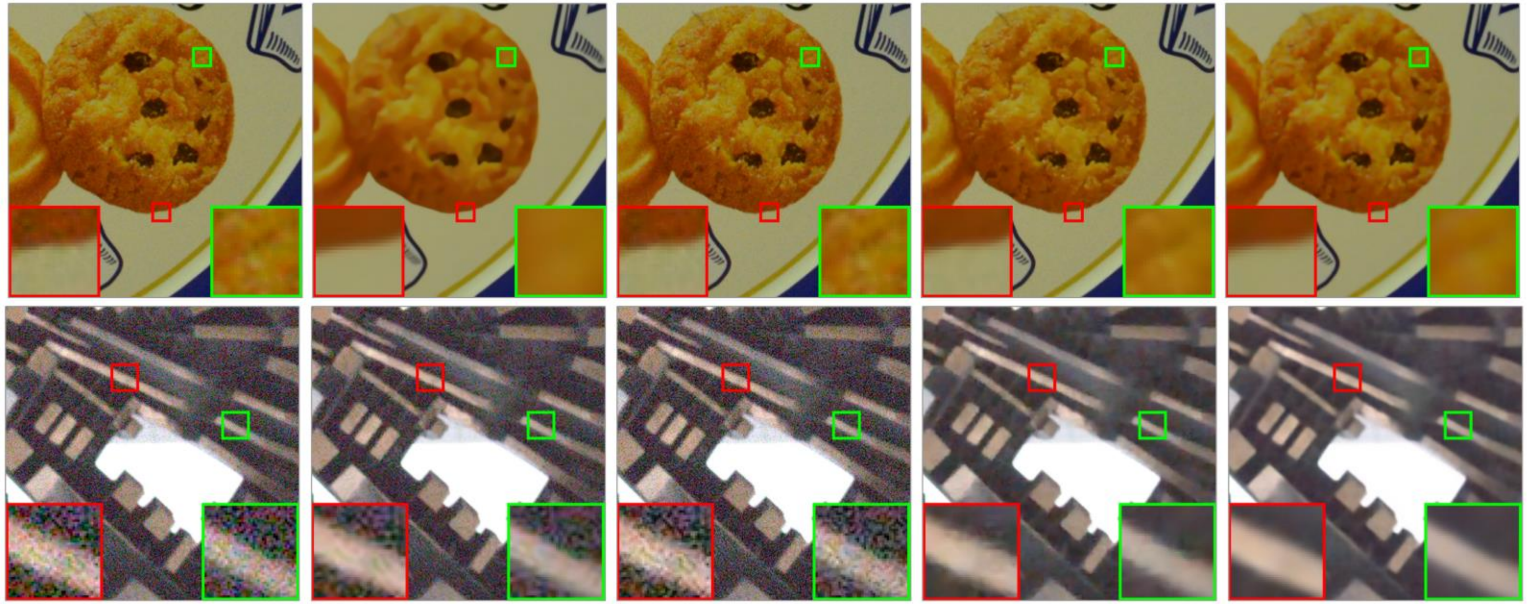
- CNN_{est}
  - y → NLF $g_y(y)$
  - Output: C*C (1 for gray level image and 3 for color image) covariance matrix for each position.
  - 1×1 convolution layers: the noise level at a position can be guaranteed to only depends on the input value at the same position
  - Each noisy image has its own network parameters in $\mathrm{CNN}_{\mathrm{est}}$ to learn image specific NLF



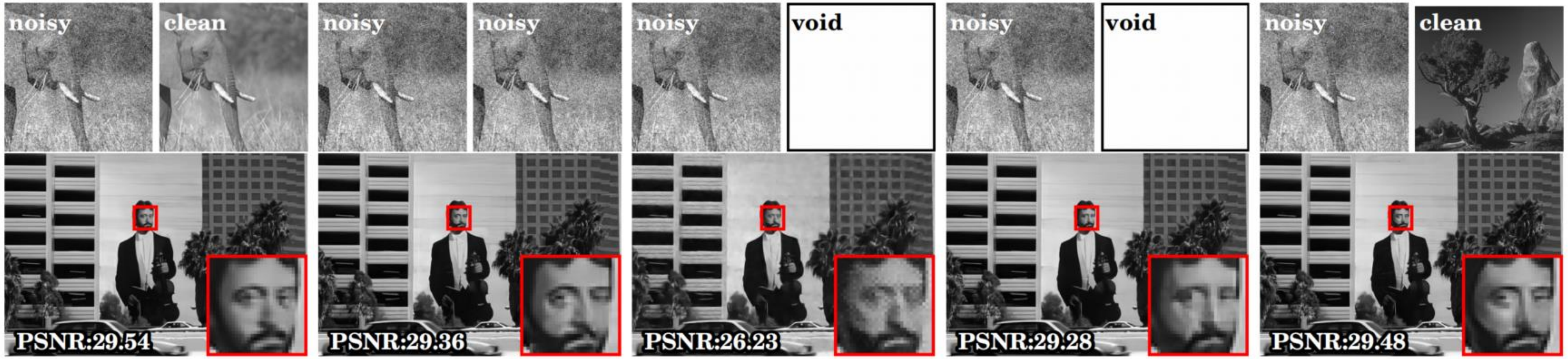
Method

- Self-Supervised Loss
  - $y_i = \tilde{x}_i + n_i, n_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i^{\mathbf{n}}); \boldsymbol{\mu} = \tilde{\mathbf{x}} + \mathbf{n}^{\boldsymbol{\mu}}, n_i^{\boldsymbol{\mu}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i^{\boldsymbol{\mu}})$
  - $\epsilon_i = y_i - \mu_i \quad \epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i^{\mathbf{n}} + \mathbf{\Sigma}_i^{\boldsymbol{\mu}})$
  - $\mathcal{L}_{self} = \sum_i \frac{1}{2} \left\{ (y_i - \hat{\mu}_i)^\top (\hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}} + \hat{\mathbf{\Sigma}}_i^{\mathbf{n}})^{-1} (y_i - \hat{\mu}_i) + \log |\hat{\mathbf{\Sigma}}_i^{\mathbf{n}}| + \mathrm{tr}\left((\hat{\mathbf{\Sigma}}_i^{\mathbf{n}})^{-1} \hat{\mathbf{\Sigma}}_i^{\boldsymbol{\mu}} \right) \right\}$

Results



Noisy BM3D [9] DnCNN [41] CBDNet [15] Ours (full)



MWCNN(N2C) [28] MWCNN(N2N) [26] N2V [22] Laine19 [24] MWCNN(unpaired)