ICCV23
PARIS

Noise2Info: Noisy Image to Information of Noise for Self-Supervised Image Denoising

Jiachuan Wang[†], Shimin Di[†], Lei Chen[†*], Charles Wang Wai Ng[†*]

[†]The Hong Kong University of Science and Technology, Hong Kong SAR, China

[*]The Hong Kong University of Science and Technology (Guangzhou), Guangdong Province, China

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

香港科技大學(廣州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

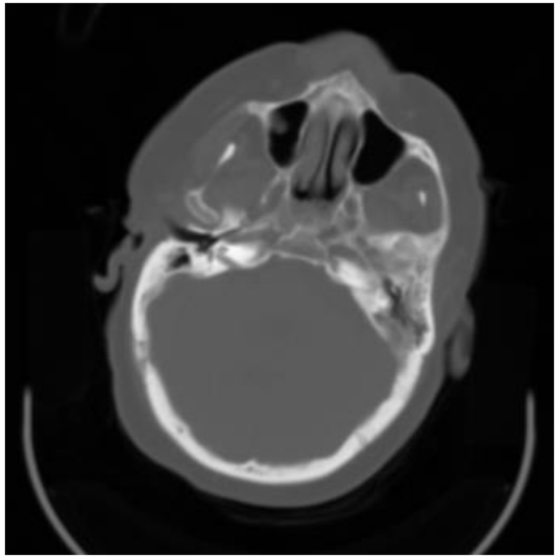
Outline

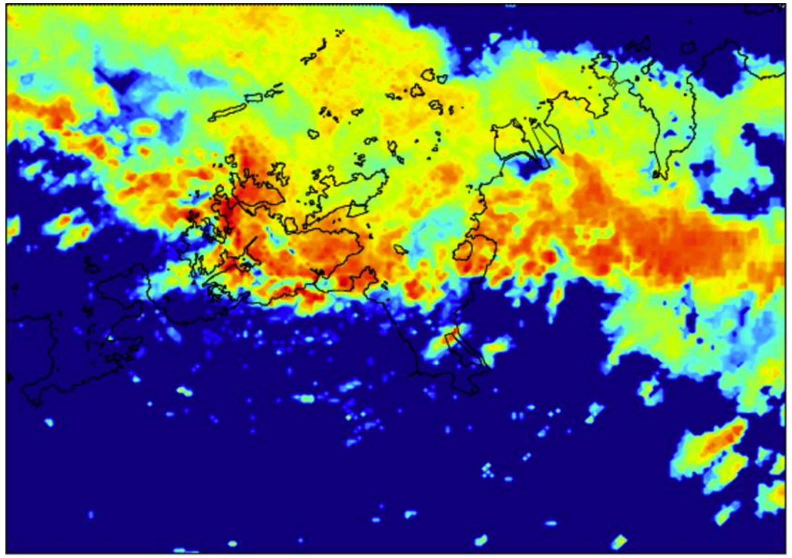
- Motivation
- Problem Formulation
- Methods
- Evaluations

Background

Image denoising is important in many scenarios

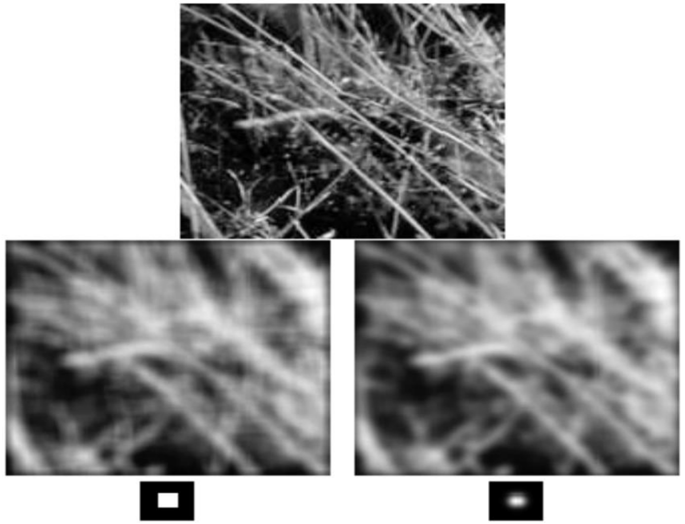
- Photo sharing on social media
- Medical images denoising
- Radar images repairing



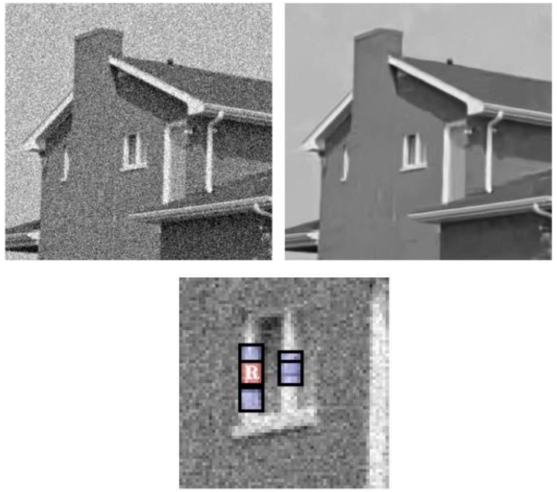


Methods for Image Denoising

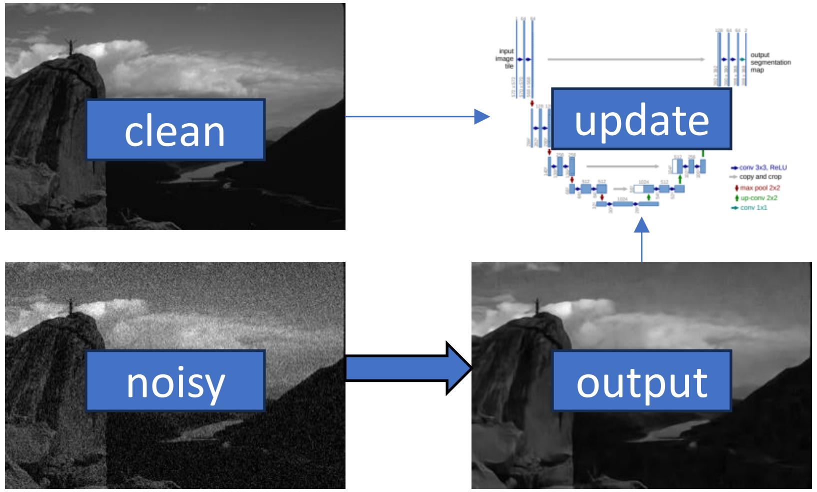
Traditional methods are quickly surpassed by deep learning methods



Filters
Smoother but blurred



BM3D
Find similar fragments

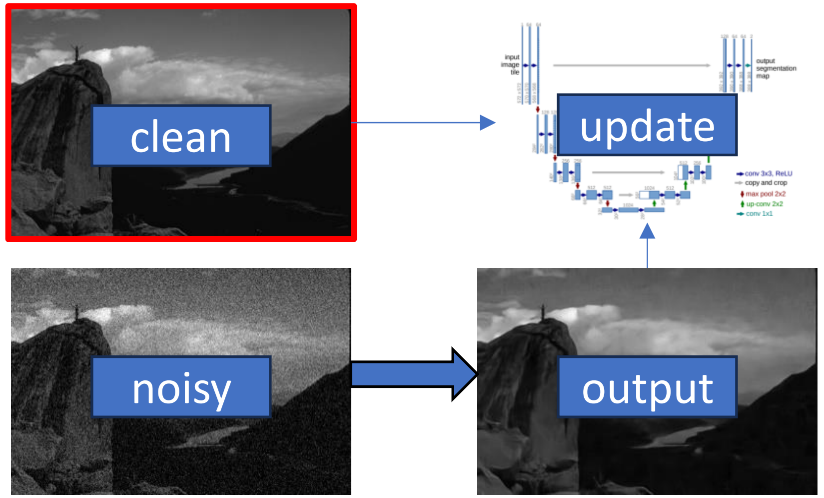


Supervised deep learning model
Loss optimization via gradient descent

Wim C. de Leeuw, Robert van Liere: BM3D: Motion Estimation in Time Dependent Volume Data. IEEE Visualization 2002

Methods for Image Denoising

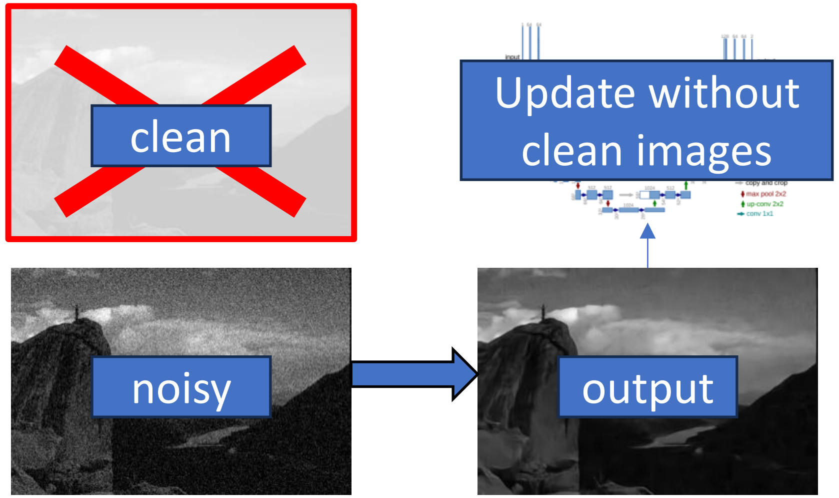
Clean images are hardly available in real-world applications!



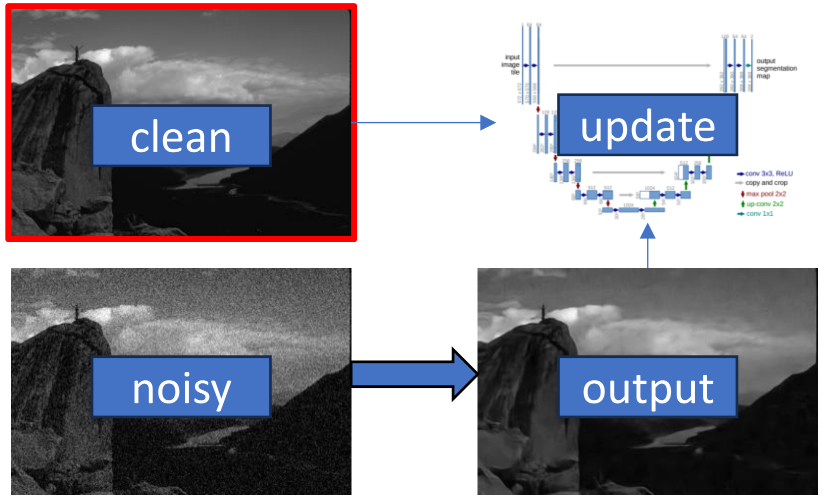
Supervised deep learning model

Loss optimization via gradient descent

Methods for Image Denoising



Clean images are hardly available in real-world applications!



Self-supervised deep learning model

Only noisy images are available

Supervised deep learning model

Loss optimization via gradient descent

Self-supervised denoising models

No clean image Y as guidance

Using noisy image X to guide training

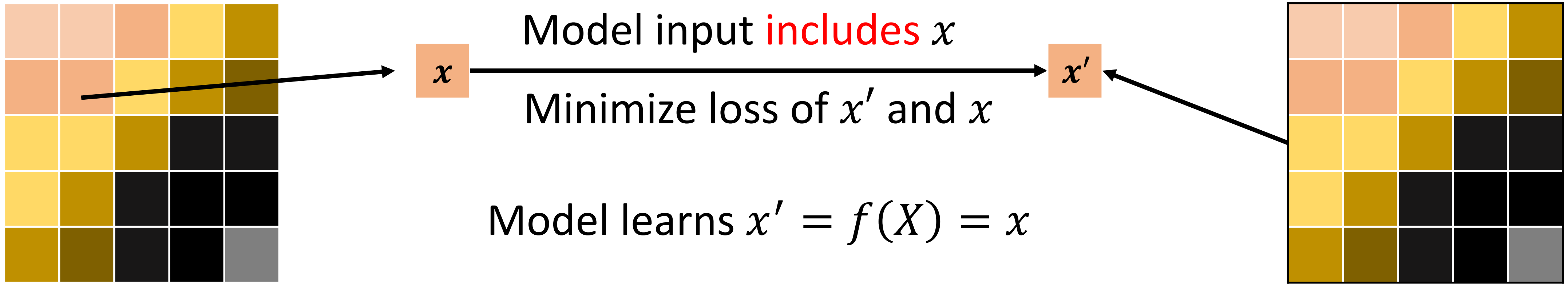
$\rightarrow$ Learn an identical function $x' = f(X) = x$

Self-supervised denoising models

No clean image Y as guidance

Using noisy image X to guide training

$\rightarrow$ Learn an identical function $x' = f(X) = x$



Model learns $x' = f(X) = x$

Noisy image X Noisy image X

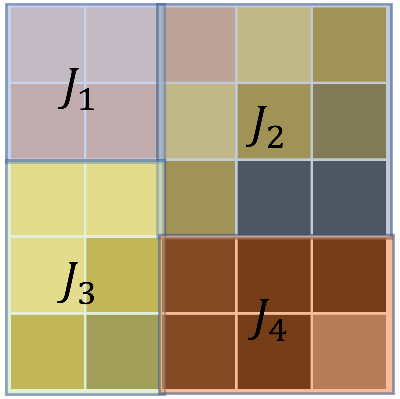
Self-supervised denoising models

Solution:

Learns each pixel x under ***J-invariant***

That is, cutting input into subsets $J = [J_1, J_2, \cdots, J_k]$

Recovering J_i only based on $J_c = J - \{J_i\}$



Noisy image X

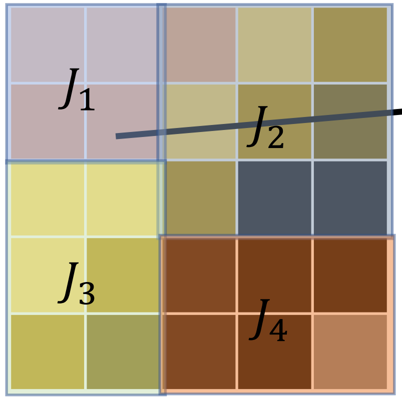
Self-supervised denoising models

Solution:

Learns each pixel x under ***J-invariant***

That is, cutting input into subsets $J = [J_1, J_2, \cdots, J_k]$

Recovering J_i only based on $J_c = J - \{J_i\}$



Noisy image X

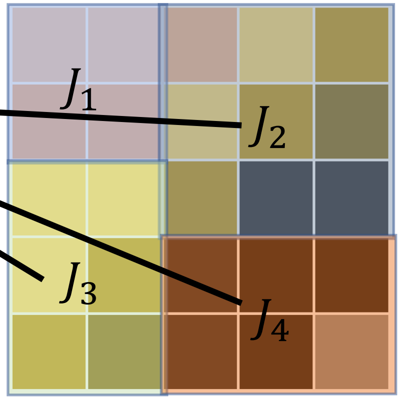
Model input excludes x

$\boldsymbol{x} \longrightarrow \boldsymbol{x'}$

Model $x' = f(J_c) = f(J_2, J_3, J_4)$

Minimize loss of x' and x

f can never learn an identical function $f(X) = x$

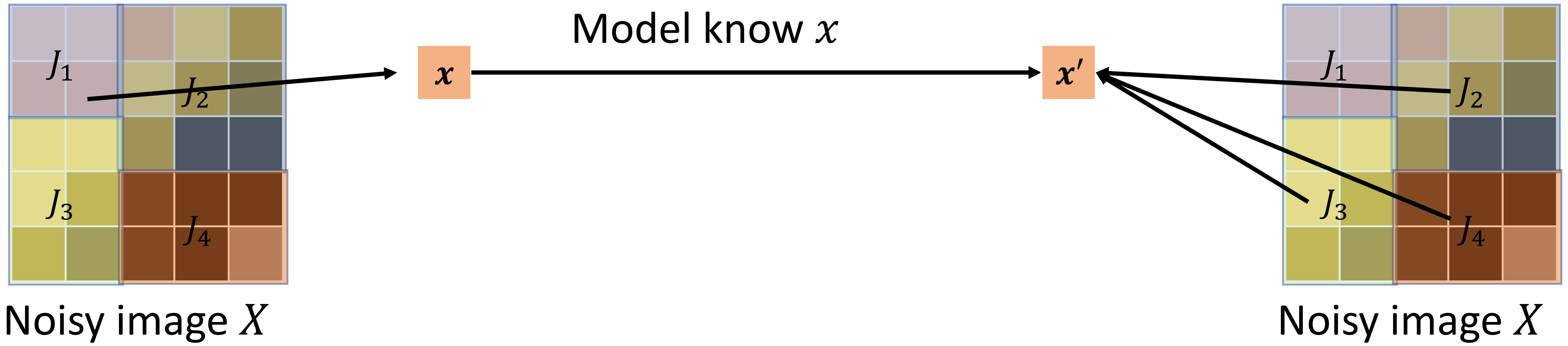


Noisy image X

Self-supervised denoising models

Problem:
Only information from J_c is used. (External information)

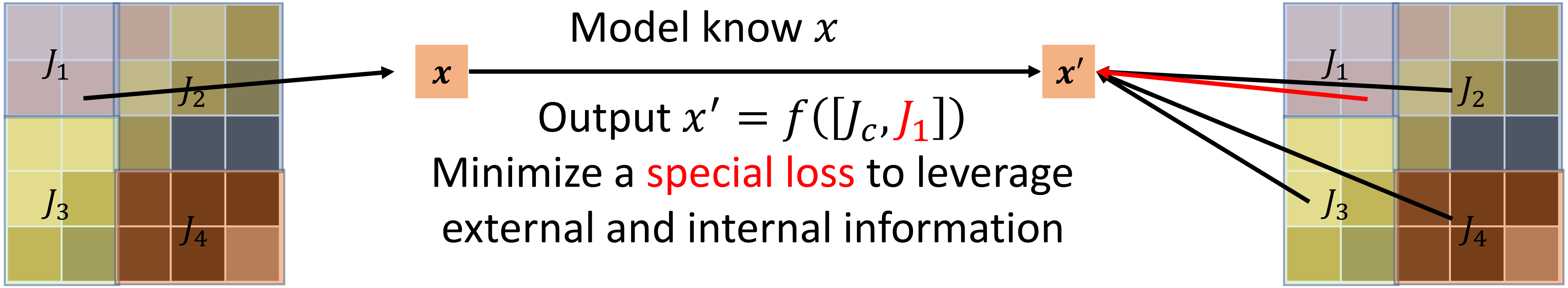
→Use information from the pixel itself. (Internal information)



Self-supervised denoising models

Problem:
Only information from J_c is used. (External information)

→Use information from the pixel itself. (Internal information)



Noisy image X Noisy image X

Self-supervised denoising models

Problem:
Only information from J_c is used. (External information)

→Use information from the pixel itself. (Internal information)

$$\mathcal{L}(\mathcal{F}, X) = \underbrace{\mathcal{L}_{in}}_{\text{Internal loss}} + 2\underbrace{\sigma_n}_{\text{The standard deviation of the noise}}\underbrace{\mathcal{L}_{ex}}_{\text{External loss}}$$

Self-supervised denoising models

Problem:

σ_n is still unknown in real-world applications.

$$\mathcal{L}(\mathcal{F}, X) = \mathcal{L}_{in} + 2\sigma_n\mathcal{L}_{ex}$$

The standard deviation of the noise

Self-supervised denoising models

Problem:
σ_n is still unknown in real-world applications.

This motivates us to design a totally self-supervised method considering both internal and external information.
→ derive σ_n-related information only using the noisy images

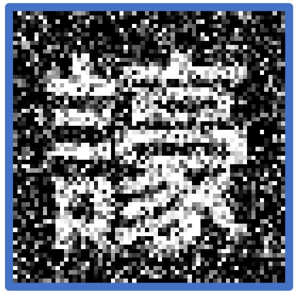
$$\mathcal{L}(\mathcal{F}, X) = \mathcal{L}_{in} + 2\sigma_n \mathcal{L}_{ex}$$

The standard deviation of the noise

Outline

- Motivation
- Problem Formulation
- Algorithms
- Evaluations

Notations



Noisy Image

$X \in \mathbb{R}^{h \times w \times c}$

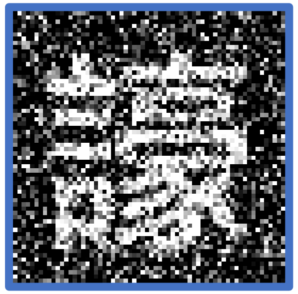
Clean Image

$Y \in \mathbb{R}^{h \times w \times c}$

$N = Y - X$

Noise Map

Supervised Learning



Noisy Image

$X \in \mathbb{R}^{h \times w \times c}$

Clean Image

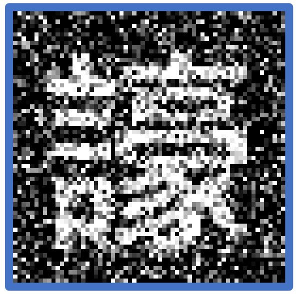
$Y \in \mathbb{R}^{h \times w \times c}$

Trained model $\mathcal{F}: \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{R}^{h \times w \times c}$

Minimize loss:

$$\mathcal{L}(\mathcal{F}, X) = \mathbb{E}_{X,Y} ||\mathcal{F}(X) - Y||$$

Self-Supervised Learning



Noisy Image

$X \in \mathbb{R}^{h \times w \times c}$

~~Clean Image~~

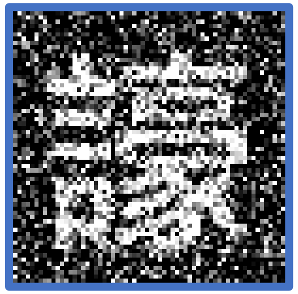
$Y \in \mathbb{R}^{h \times w \times c}$

Trained model $\mathcal{F}: \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{R}^{h \times w \times c}$

Minimize loss:

$$\mathcal{L}(\mathcal{F}, X) = \mathbb{E}_X ||\mathcal{F}\left(X_{J_c}\right) - X_{J_i}||$$

Self-Supervised Learning



Noisy Image

$X \in \mathbb{R}^{h \times w \times c}$

Train model $\mathcal{F}$ to consider internal information (X_{J_i})

model $\mathcal{F}$ only considers external information (X_{J_c})

$$\mathcal{L}(\mathcal{F}, X) = \mathbb{E}_X || \boxed{\mathcal{F}(X_{J_c})} - X_{J_i} ||$$

Self-Supervised Learning

Train model $\mathcal{F}$ to consider internal information

Supervised loss Constant

$$\mathcal{L}(\mathcal{F}, X) = \mathbb{E}_{X,Y}\left[\boxed{||\mathcal{F}(X_{J_c}, X_{J_i}) - Y||^2} + \boxed{||X - Y||^2}\right]$$

$$\leq \mathbb{E}_X \left|| \mathcal{F}(X) - X \right||^2$$

$$+2\sigma_n \cdot m\mathbb{E}_J \left[\mathbb{E}||\mathcal{F}(X)_J - \mathcal{F}(X_{J_c})_J||^2/|J|\right]^{1/2}$$

$$= \mathcal{L}_{in} + 2\sigma_n \mathcal{L}_{ex}$$

$\mathcal{L}_{in}$ and $\mathcal{L}_{ex}$ only need noisy image,

but σ_n still needs information of noise

Self-Supervised Learning

Train model $\mathcal{F}$ to consider internal information

Problem

How to estimate σ_n only based on noisy images, so that the method is **end-to-end self-supervised**?

$\mathcal{L}_{in}$ and $\mathcal{L}_{ex}$ only need images image,
but σ_n still needs information of noise

Outline

- Motivation
- Problem Formulation
- Algorithms
- Evaluations

Proposed Approaches

1. A theoretical upper bound

We first propose a bound for σ_n

$$\sigma_n \leq \frac{\mathcal{L}_{ex} + \sqrt{\mathcal{L}_{ex}^2 + m\left(\mathcal{L}_{in} - E_{X,Y}[||\mathcal{F}(X) - \boxed{\boldsymbol{Y}||^2}]\right)}}{m},$$

where all the terms are tractable except $\boldsymbol{Y}$,

which is the clean image.

Proposed Approaches

1. A theoretical upper bound

We first propose a bound for σ_n

$$\sigma_n \le \frac{\mathcal{L}_{ex} + \sqrt{\mathcal{L}_{ex}^2 + m\left(\mathcal{L}_{in} - \boxed{E_{X,Y}[||\mathcal{F}(X) - \boxed{\boldsymbol{Y}}||^2]}\right)}}{m},$$

where all the terms are tractable except Y, which is the clean image.

2. Transformation

We further transform the intractable term:

$$E_{X,Y}[||\mathcal{F}(X) - \boldsymbol{Y}||^2]$$

$$= E_{X,Y}\left[\left|\left|(X - \boldsymbol{Y}) - \left(X - \mathcal{F}(X)\right)\right|\right|^2\right]$$

$$= E_{X,N}[||\boxed{\boldsymbol{N}} - \boxed{\widetilde{N}(X)}||^2]$$

real noise (intractable)

"noise" removed by our model (tractable)

Proposed Approaches

1. A theoretical upper bound

2. Transformation

We further transform the intractable term:

$$E_{X,Y}[||\mathcal{F}(X) - \boldsymbol{Y}||^2]$$

$$= E_{X,Y}[||(X - \boldsymbol{Y}) - (X - \mathcal{F}(X))||^2]$$

$$= E_{X,N}[||\boxed{\boldsymbol{N}} - \boxed{\widetilde{N}(X)}||^2]$$

real noise (intractable)

"noise" removed by our model (tractable)

3. Transfer to tractable distribution

As the distribution of noise $\boldsymbol{N}$ is unknown, we use the maximum likelihood estimation (**MLE**) $\mathcal{N}^*$ of removed noise to get a smaller estimation of the original term.

$$E_{X,N^*}[||\boxed{\boldsymbol{N}^*} - \widetilde{N}(X)||^2]$$

Using $\mathcal{N}^*$, we can sample N^* for estimation.

Proposed Approaches

1. A theoretical upper bound

2. Transformation

3. Transfer to tractable distribution

As the distribution of noise N is unknown, we use the maximum likelihood estimation (**MLE**) $\mathcal{N}^*$ of removed noise to get a smaller estimation of the original term.

$$E_{X,N^*}\left[||N^* - \widetilde{N}(X)||^2\right]$$

Using $\mathcal{N}^*$, we can sample N^* for estimation.

The derived **MLE** of removed noise is shown in the following lemma

Lemma 2 (MLE of samples from $\tilde{\mathbf{n}}$). *We denote the maximum likelihood estimation of $\tilde{\mathbf{n}}$ as $n^* \sim \mathcal{N}^*$, which has distribution:*

$$P(n^* = \tilde{N}_j^{(i)}) = (mq)^{-1} \qquad \forall \tilde{N}_j^{(i)} \in \tilde{\mathbf{n}}, \tag{8}$$

where $\tilde{N}_j^{(i)}$ represents $\tilde{N}(X^{(i)})_j$ for short.

Proposed Approaches

1. A theoretical upper bound

2. Transformation

3. Transfer to tractable distribution

As the distribution of noise $\boldsymbol{N}$ is unknown, we use the maximum likelihood estimation (**MLE**) $\mathcal{N}^*$ of removed noise to get a smaller estimation of the original term.

$$E_{X,N^*}[||\boldsymbol{N}^* - \widetilde{N}(X)||^2]$$

Using $\mathcal{N}^*$, we can sample N^* for estimation.

4. Relaxation to tractable estimation

In each batch, the k sampled pixels N^* are **index-free**. How they map to the k pixels in $\widetilde{N}(X)$ is still unknown.

We derive a **tractable** bound, which is the optimal case when model is well-trained.

$$\geq E_{N^*}\left[E_X\left[\sum_{j=1}^{m}(N^*_{v_j} - \tilde{N}(X)_{u_j})^2\right]\right]$$

Proposed Approaches

1. A theoretical upper bound

2. Transformation

3. Transfer to tractable distribution

The bound is given in the following lemma

Lemma 3. *Given the sampled noise map N^* from $\mathcal{N}^*$, we sort the m pixels of the removed noise map $\tilde{N}(X)$ ($\{\tilde{N}(X)_j\}_{j=1}^m$) in increasing order and define the index list as $\{u_1, u_2, \cdots, u_m\}$, i.e., $\tilde{N}(X)_{u_1} \leq \tilde{N}(X)_{u_2} \leq \tilde{N}(X)_{u_3} \leq \cdots \leq \tilde{N}(X)_{u_m}$. Similarly, we define the index list for increasingly sorted sampled noise pixels $\{N_j^*\}_{j=1}^m$ as $\{v_1, v_2, \cdots, v_m\}$. We have:*

$$\mathbb{E}_{N^*}[\mathbb{E}_X[\sum_{j=1}^{m}(N_j^* - \tilde{N}(X)_j)^2]]$$
$$\geq \mathbb{E}_{N^*}[\mathbb{E}_X[\sum_{j=1}^{m}(N_{v_j}^* - \tilde{N}(X)_{u_j})^2]]. \tag{10}$$

4. Relaxation to tractable estimation

In each batch, the k sampled pixels N^* are **index-free**. How they map to the k pixels in $\widetilde{N}(X)$ is still unknown.

We derive a **tractable** bound, which is the optimal case when model is well-trained.

$$\geq E_{N^*}\left[E_X\left[\sum_{j=1}^{m}(N_{v_j}^* - \tilde{N}(X)_{u_j})^2\right]\right]$$

Proposed Approaches

1. A theoretical upper bound

2. Transformation

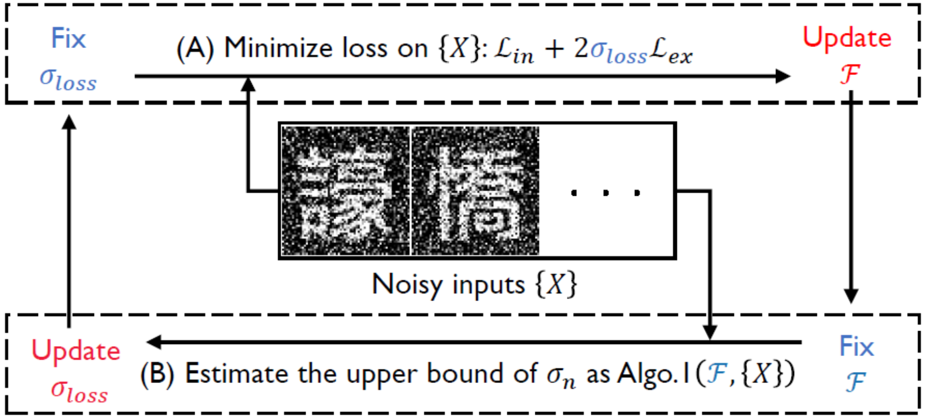
3. Transfer to tractable distribution

4. Relaxation to tractable estimation

5. Training and updating algorithm

An overall training algorithm ***Noise2Info***.

Update model and estimate σ_n alternatively



Algorithm 2 Noise2Info

Input: The denoising model $\mathcal{F}$, noisy images $\mathbf{X} = \{X^{(i)}\}_{i=1}^{p}$, the number of epochs k_r, the number of samples for model updation k_t and σ_n estimation k_u.
Initialize $\sigma_{loss} \leftarrow 1$.
for $i \leftarrow 1$ **to** k_r **do**
  Update $\mathcal{F}$ via loss $\mathcal{L}_{in} + 2\sigma_{loss}\mathcal{L}_{ex}$ with k_t samples.
  $\sigma^*_{loss} \leftarrow$ Algorithm 1($\mathcal{F}$, k_u noisy images, k_{mc})
  if $\sigma^*_{loss} < \sigma_{loss}$ **then**
    $\sigma_{loss} \leftarrow \sigma^*_{loss}$
  end if
end for
Return: model $\mathcal{F}$ for denoising

Outline

- Motivation
- Problem Formulation
- Algorithms
- Evaluations

Experimental Setting

- Benchmark Datasets (Self-supervised denoising)
  - ImageNet
  - Hanzi
  - BSD68
- Real-World Datasets
  - SIDD
  - PolyU
- Synthetic Datasets
  - Inject different scales of noise
  - Inject different types of noise

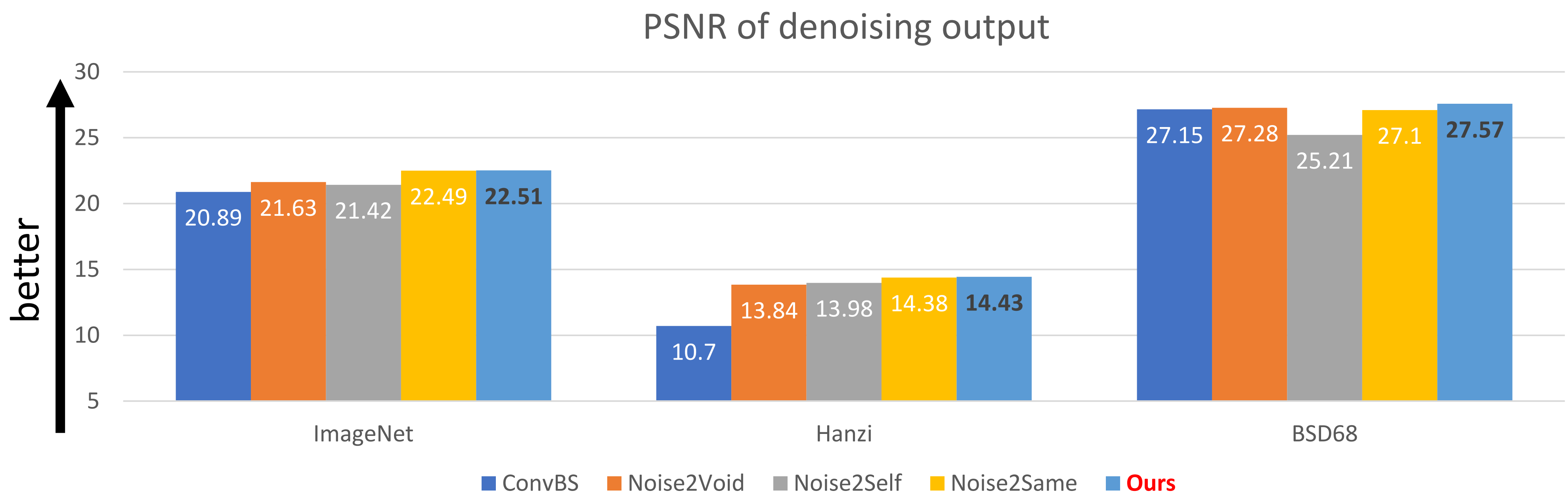
Experimental Setting

Tested Algorithms

- Traditional methods
  - NLM
  - BM3D
- Supervised methods
  - Noise2True
  - Noise2Noise
- Self-supervised methods
  - Noise2Void
  - Noise2Self
  - ConvBS
  - Noise2Same

Experimental Results

Metric: Peak Signal-to-Noise Ratio (PSNR) evaluates the similarity between two images.
The **larger** the better.



Performance of self-supervised methods with donut masking.
Our Noise2Info **outperforms** other methods on 3 benchmarks.

Experimental Results

Metric: Peak Signal-to-Noise Ratio (PSNR) evaluates the similarity between two images.
The **larger** the better.

Table 5: The performance on the Hànzì dataset on more noise types. N2S denotes Noise2Same ($\sigma_{loss} = \sigma_n$). FBI [6] is a denoising method designed for Poisson-Gaussian noise.

Model	Types of injected noises					
	Poisson-Gaussian (A) $\sigma_n = 0.8181, \mu = 0.0002$		Poisson-Gaussian (B) $\sigma_n = 10.32, \mu = 6.34$		Pepper $\sigma_n = 0.8492, \mu = 0.3037$	
	PSNR	σ_{loss}	PSNR	σ_{loss}	PSNR	σ_{loss}
Noise2Void	18.88	-	17.93	-	23.77	-
Noise2Self	18.91	-	17.57	-	22.19	-
Noise2Same	18.91	0.8181	14.49	10.32	24.35	0.8492
FBI [6]	18.87	N/A	6.54	N/A	N/A	N/A
Noise2Info	**19.11**	0.8317	**18.52**	0.8551	**24.96**	0.7043

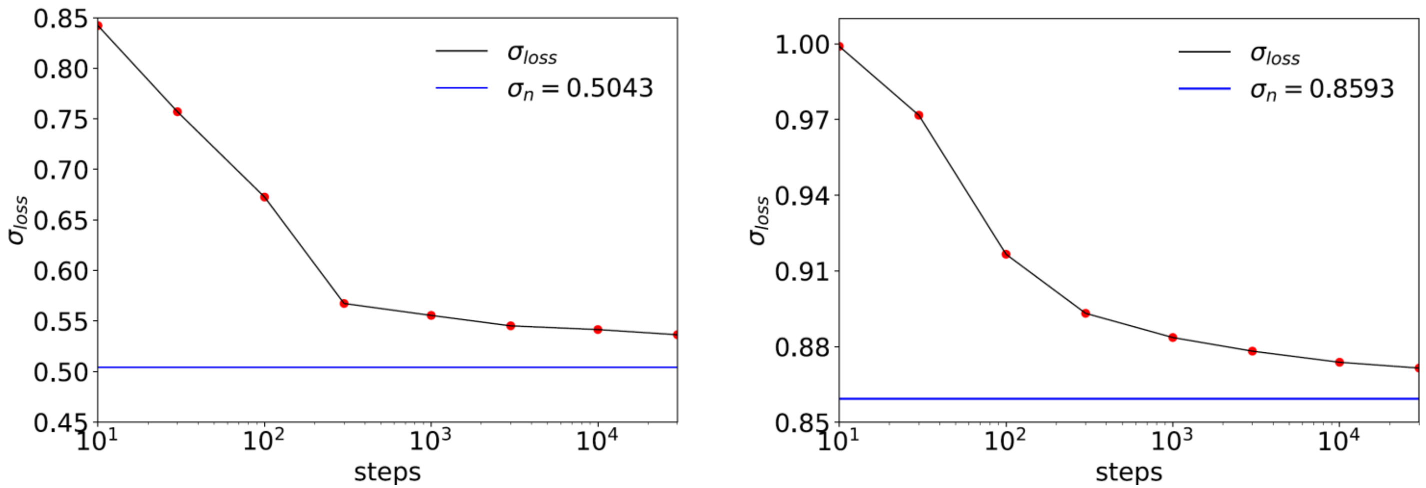
Performance of self-supervised methods on 2 types of noise out of our theory assumption (zero-mean and signal independent). Our Noise2Info **outperforms** other methods on 3 types of noise.

Experimental Results

Estimation and real σ_n on Hanzi dataset

Estimation σ_{loss}	0.6006	0.7818	0.8710	0.9187
Real std σ_n	0.5845	0.7683	0.8593	0.9075

Our estimation σ_{loss} **closely upper bound** the real σ_n



(a) Hànzì data set.

(b) BSD68 data set.

σ_{loss} estimation in different training steps.

The estimation σ_{loss} gets closer to σ_n as training steps increase

Thank You

Q&A

The code and datasets
https://github.com/dominatorX/Noise2Info-code