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
jwangey@connect.ust.hk, sdiaa@connect.ust.hk, leichen@hkust-gz.edu.cn, charles.ng@ust.hk

Abstract

Unsupervised image denoising has been proposed to alleviate the widespread noise problem without requiring clean images. Existing works mainly follow the self-supervised way, which tries to reconstruct each pixel \( x \) of noisy images without the knowledge of \( x \). More recently, some pioneer works further emphasize the importance of \( x \) and propose to weigh the information extracted from \( x \) and other pixels when recovering \( x \). However, such a method is highly sensitive to the standard deviation \( \sigma_n \) of noise injected to clean images, where \( \sigma_n \) is inaccessible without knowing clean images. Thus, it is unrealistic to assume that \( \sigma_n \) is known for pursuing high model performance.

To alleviate this issue, we propose Noise2Info to extract the critical information, the standard deviation \( \sigma_n \) of injected noise, only based on the noisy images. Specifically, we first theoretically provide an upper bound on \( \sigma_n \), while the bound requires clean images. Then, we propose a novel method to estimate the bound of \( \sigma_n \) by only using noisy images. Besides, we prove that the difference between our estimation with the true deviation goes smaller as the model training. Empirical studies show that Noise2Info is effective and robust on benchmark data sets and closely estimates the standard deviation of noise during model training.

1. Introduction

Generally, images are vulnerable to noise from latent observation and transmission [5, 26]. As an essential enhancement for digital images, image denoising aims to convert noisy images \( X \) to clean ones \( Y \), where the image denoising model \( F(X) \) is expected to output the near clean image (i.e., \( F(X) \approx Y \)). Assuming that clean images are available, deep learning models [23, 16] have been introduced to the image denoising task, and achieved outstanding performance over traditional methods [10, 11, 12] on the supervised image denoising task [31, 22, 14].

However, in real-world scenarios, only noisy images can be observed, i.e., we do not know whether an image has been contaminated or what the ground truth of a noisy image (a clean one) looks like. Thus, it is hard to apply the supervised deep learning approaches. To handle such cases, Noise2Noise [19] assumes that pairwise noisy images of one clean image can be accessible, which can be viewed as noisy supervision [29]. On the other hand, many papers assume that the distribution of noise is known, named noise model. A common setting for noise \( n \sim \mathcal{N} \) is zero-mean (\( \mu_n = 0 \)) with unknown standard deviation (\( \sigma_n \)) [17, 18, 29, 3]. The Gaussian and multiplicative Bernoulli noises have also been covered [21]. CBDNet [14] assumes that photographs have Poisson-Gaussian noise. However, their assumptions are not always held in reality and limit the applicability of their methods.

To enable models to denoise on a more practical scenario, recent works (e.g., Noise2Self [3], Noise2Void [17] and Convolutional blind-spot network [18]) develop self-supervised models mainly based on available noisy images. If a model \( F \) takes noisy image \( X \) as both input and target, it will quickly collapse to the identity function \( F(X) = X \). Instead, these papers use the idea of \( \mathcal{J} \)-invariance [3]. Loosely speaking, given noise image \( X \), a \( \mathcal{J} \)-invariant model denoises each pixel \( x \in X \) only based on any other pixels (i.e., using pixels from \( X \backslash \{ x \} \)). This setting prevents model from learning the identity function. Though the pixels used for supervision are noisy, with many samples drawn from the same image distribution, the model is supposed to learn the expected ground truth value.

For the strictly \( \mathcal{J} \)-invariant model, each pixel is denoised without using the pixel itself, so that we call it external method. Based on the idea, many papers point out that the extracted information based on the pixel itself, which is the internal information of the pixel, can be utilized for better results [3, 18, 29]. Noise2Same [29] considers both the external and internal information to further outperform these purely external models. Formally, Noise2Same builds a self-supervised \( \mathcal{L}(F, X) \) (i.e., the loss of \( F \) w.r.t. \( X \)) on
the top of internal loss $\mathcal{L}_{in}$ and external loss $\mathcal{L}_{ex}$ as:

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

which is an upper bound of the typical supervised loss. For a normalized noisy image, $\sigma_{loss}$ is proved to be the standard deviation of its noise $\sigma_n$. As $\sigma_n$ is not available, Noise2Same uses $\sigma_{loss} = 1$ by default instead. Note that the closer $\sigma_{loss}$ and $\sigma_n$ are, the better the image denoising performance. As shown in Fig. 1, we show an image (Fig. 1 (a)) from Hanzì dataset and its noisy version (Fig. 1 (b)) with std of noise $\sigma_n = 0.7683$. When $\sigma_{loss}$ is set to 0.5 and 1 (Fig. 1(c) and Fig. 1 (e)), the performances of Noise2Same are not so desirable compared with that of $\sigma_{loss} = \sigma_n$ (Fig. 1 (d)). However, $\sigma_n$ can only be known when clean images are available, which contradicts the purpose of practical image denoising. Thus, it is hard to manually set $\sigma_{loss}$ close to $\sigma_n$ for better performance, especially when there is no clean image for tuning $\sigma_{loss}$. Besides, as shown in Fig. 4 of [29], the quality of denoised images is highly sensitive to $\sigma_{loss}$.

In this paper, we aim to solve the above issue to enable the image denoising model to work well when no clean image nor noise model is available. Motivated by the observation on $\sigma_n$, we propose Noise2Info to derive $\sigma_n$-related information by only taking the noisy images as inputs, which has not been studied in existing works. First, we theoretically estimate the upper bound of $\sigma_n$ in Noise2Info. Then, based on the estimation, Noise2Info can dynamically update $\sigma_{loss}$ during the model training. In addition, we prove that the gap between estimated upper bound with the true standard deviation will become smaller as model training, leading to a convergent stable result. The empirical study shows that Noise2Info outperforms other self-supervised methods and achieves comparable results over the supervised methods. Especially, Noise2Info even beats all self-supervised methods including Noise2Same with known $\sigma_n$ on the two benchmark data sets where noises are signal-dependent and not zero-mean, which validates the generality of our method. We also synthesize data sets with various noise types and scales. As shown in Tab. 4 and Tab. 9, the gap between $\sigma_n$ and $\sigma_{loss}$ estimated by Noise2Info is pretty small ($< 0.02$), which verifies that Noise2Info can indeed estimate $\sigma_n$ only based on noisy images.

**Notations.** In this paper, we denote the lower case $a$ to the scalar and the upper case $A \in \mathbb{R}^m$ to the vector. We use the superscript $A(\cdot) \in \mathbb{R}^m$ to denote the $i$-th sample, the bold font $\mathbf{A}$ to the set, like $\mathbf{a} = \{a^{(1)}, \ldots, a^{(n)}\}$ and $\mathbf{A} = \{A^{(1)}, \ldots, A^{(q)}\}$. The subscript $A_j^{(\cdot)}$ is $j$-th element of $A^{(\cdot)}$. The Fraktur case $\mathcal{A}$ denotes the function. Besides, $\tilde{A}$ denotes the output of the denoising model, and the star $A^*$ denotes the estimation.

2. Background and Related Works

2.1. Image denoising

We categorize various image denoising models based on the form of supervision below.

**Unsupervised Denoising.** Many traditional methods based on the assumptions of smoothness and self-similarity of the image fall into this category. These models do not
need to be trained and thus have a wide range of applicabilities but their performance is unstable [5, 26]. Various filters can be viewed as denoisers, such as Mean filter, Median filter, and Gauss filter [13]. The non-local means algorithm [4] is proposed as a more powerful mean filter. It outputs the mean of weighted pixels from the whole image instead of neighbors, where the weights are set according to similarity. BM3D algorithm [10] further improves the results. Similar image fragments are grouped and stacked as blocks. These blocks are further transferred to frequency space and applied with thresholding to filter high frequency noise. BM3D has many hyperparameters including the standard deviation of noise for thresholding, which limits its applicability for blind denoising.

**Denoising with Paired Input and Target.** The image denoising can be viewed as a general regression task with paired noisy and clean images \((X, Y)\), where \(X, Y \in \mathbb{R}^m\) and \(m = h \times w \times c\) is the number of pixels of each RGB image. The noisy image can be viewed as a combination of the clean image and a noise map \(N = X + N\), where \(N \in \mathbb{R}^m\) and the pixels of \(N\) are i.i.d. The denoising model \(\mathcal{F} : \mathbb{R}^m \rightarrow \mathbb{R}^m\) aims to minimize the loss function:

\[
\mathcal{L} = \mathbb{E}_{X, Y} ||\mathcal{F}(X) - Y||, \tag{2}
\]

where \(|| \cdot ||\) is the distance metric. DnCNN [31] learns the residual noise map \(\tilde{N} \in \mathbb{R}^m\) and uses \(X - \tilde{N}\) as the final output. CBDNet [14] assumes that the noise map \(\tilde{N}\) is more likely to follow the mixed Poisson-Gaussian distribution.

However, clean images are usually not available in real-world scenarios. Noise2Noise [19] builds the denoising model with noisy pairs \((X_1, X_2)\), which has loss function:

\[
\mathcal{L} = \mathbb{E}_{X_1, X_2} ||\mathcal{F}(X_1) - X_2||. \tag{3}
\]

As long as the pair of noisy samples have zero-mean and independent noise, Noise2Noise proves that training on paired noisy images is the same as training on noisy and clean images because of the proof \(\mathbb{E}[X_2|X_1] = Y\). Under such noisy supervision, Noise2Noise even outperforms the supervised models trained on clean images of some datasets:

**Self-supervised Denoising.** It is still hard to hold the assumption of Noise2Noise that two or more samples with independent noise for a clean image exist. Noise2Self [3] first proposes the \(\mathcal{J}\)-invariance to handle a more general and realistic case, which trains denoising models only with one noisy observation, i.e., self-supervised image denoising.

**Definition 1 (\(\mathcal{J}\)-invariance).** Given a noisy image \(X \in \mathbb{R}^m\), let \(\mathcal{J} = \{J^{(1)}, J^{(2)}, \cdots, J^{(k)}\}\) be the non-intersecting partition of image \(X\), and \(X_j\) be the pixels in the partition \(J \in \mathcal{J}\), i.e., \(\text{concatenate}(X_{J^{(1)}}, \ldots, X_{J^{(k)}}) = X\). Then, let \(J^c = \mathcal{J}\setminus\{J\}\) denote the complement of \(J\). The \(\mathcal{J}\)-invariant function is defined as:

\[
[\mathcal{F}(X_{J^c})]_J = [\mathcal{F}(X)]_J, \forall J \in \mathcal{J}.
\]

The \(\mathcal{J}\)-invariant function tries to recover pixels in partition \(J\) by only using information from other partitions \(X_{J^c}\), which can be regarded as self-supervision. It can force the model to extract the correlation between the pixels of one partition with those of other partitions.

Based on the \(\mathcal{J}\)-invariance, the \(\mathcal{J}\)-invariant models Noise2Self [3], Noise2Void [17], and ConvBS [18] propose to minimize losses of the form:

\[
\mathcal{L} = \mathbb{E}_X[\mathcal{L}(\mathcal{F}, X)], \tag{4}
\]

where the model \(\mathcal{F}\) is updated based only on each noisy image \(X\). Generally, the pixels in clean images are highly correlated. Assuming that the noises in every pixel are independent, such \(\mathcal{J}\)-invariant models can eliminate the noise of a given pixel by leveraging the neighbor information of this pixel. To exclude the information of pixel itself, Noise2Self and Noise2Void use masks while ConvBS designs a special convolution layer with restricted receptive field.

**2.2. Information exploration and Bound of Noise2Same**

\(\mathcal{J}\)-invariant models only make use of external information, i.e., only \(X_{J^c}\) without \(X_J\). Many papers point out that these models clearly waste the internal information, i.e., the information of the pixel itself \(X_J\), which can further improve the result [3, 18, 29]. Noise2Self [3] states that a linear combination of \(X_J\) and \(\mathcal{F}(X_{J^c})\) improves the result when the standard deviation of noise \(\sigma_n\) is known. ConvBS [18] can improve the result with post-processing if the noise model is known. The advanced Noise2Same [29] proposes a theoretical bound over the loss in Eq. (2), which combines external and internal information:

\[
\begin{align*}
\mathbb{E}_{X, Y} & \left[ ||\mathcal{F}(X) - Y||^2 + ||X - Y||^2 \right] \\
& \leq \mathbb{E}_X ||\mathcal{F}(X) - X||^2 \\
& \quad + 2\sigma_n \cdot m \mathbb{E}_{J} \left[ ||\mathcal{F}(X)_J - \mathcal{F}(X_{J^c})_J||^2 / |J| \right]^{1/2} \\
& = \mathcal{L}_{in} + 2\sigma_n \cdot \mathcal{L}_{ex}. \tag{5}
\end{align*}
\]

We denote components on the left and right of Eq. (5) as \(\mathcal{L}_{in}\) and \(\mathcal{L}_{ex}\). \(\mathcal{L}_{in}\) pushes the output \(\mathcal{F}(X)\) similar to the noisy input \(X\) itself. \(\mathcal{L}_{ex}\) leads to the output that depends more on \(X_{J^c}\) by restricting \(\mathcal{F}(X)_J - \mathcal{F}(X_{J^c})_J\). However, without the information of standard deviation of noise \(\sigma_n\), Noise2Same can only use \(\sigma_{loss} = 1\) instead of \(\sigma_n\). In this paper, we aim to estimate \(\sigma_n\) based on the noisy images.

**3. Noise2Info**

In this section, we first introduce a tractable estimation on the upper bound of \(\sigma_n\). We analyze the tightness of the bound and demonstrate that the bound could be tighter during model training. Then, we utilize the estimated bound as \(\sigma_{loss}\) in Eq. 1, and introduce how we train the model.
need to estimate the lower bound of classic loss in Sec.
only noisy image clean image for each noisy image are functions of
where
in Lemma 1
in Lemma
in Eq.
\mathcal{L}(X,Y) \sim \mathcal{N} \ 	ext{i.i.d among all the dimensions, the standard deviation of noise} \ \sigma_n \ \text{can be upper bounded to:}
\sigma_n \leq (\mathcal{L}_{ex} + \sqrt{\mathcal{L}_{ex}^2 + m(\mathcal{L}_{in} - \mathbb{E}_{X,Y}[||F(X) - Y||^2])}/m,
\text{where} \ m = \text{the dimension of each image and} \ \mathcal{L}_{ex} \text{and} \ \mathcal{L}_{in} \ \text{are functions of} \ X \text{and} \ F \ \text{introduced in Sec. 2.} \ Y \ \text{is the clean image for each noisy image} \ X.

In above lemma, \mathcal{L}_{ex} \text{and} \ \mathcal{L}_{in} \ \text{are tractable with}
only noisy image \ X \ \text{and Eq. 5). However, term}
\mathbb{E}_{X,Y}[||F(X) - Y||^2] \ \text{(i.e.,} \ \mathcal{L} \ \text{in Eq. 2) requires clean}
image \ Y, \ \text{which is usually inaccessible as discussed in Sec. 2.1. To achieve an upper bound of} \ \sigma_n, \ \text{we need to estimate the lower bound of classic loss} \ \mathcal{L} = \mathbb{E}_{X,Y}[||F(X) - Y||^2]. \ \text{We first rewrite} \ \mathcal{L} \ \text{as:}
\mathbb{E}_{X,Y}[||X - Y||^2] = \mathbb{E}_{X,N}[||N - \tilde{N}(X)||^2],
(7)
where \ N = X - Y \ \text{is the true injected noise and} \ \tilde{N}(X) = X - F(X) \ \text{is the noise removed by the denoising model} \ F. \ \text{Note that} \ \tilde{N}(X) \ \text{can be observed for any given} \ F, \ \text{while} \ N \ \text{is unknown due to the inaccessible clean image} \ Y. \ \text{We next introduce the motivational ideas of how to estimate Eq. 7.}

As mentioned before, existing works assume that the noises contained in images are i.i.d. for any given dataset \cite{17, 18, 29, 3}. Here we demonstrate several examples from the Hánzi \cite{3} dataset in Fig. 2. Given 3 noisy images \{X^{(i)}\}_{i=1}^3 \text{ in Fig. 2 (a) and their corresponding clean images} \ \{Y^{(i)}\}_{i=1}^3 \text{ in Fig. 2 (b), the true noise maps} \ \{N^{(i)}\}_{i=1}^3 = \{X^{(i)} - Y^{(i)}\}_{i=1}^3 \ \text{are shown in Fig. 2 (e). Then, we plot the cumulative distribution function (CDF) of each noise map} \ N \ \text{and the maximum likelihood estimation (MLE) of them in Fig. 2 (h). It is clear that they have similar patterns, i.e., follow the same real distribution (red line). Motivated by the assumption and empirical observation, we propose the idea that} X - F(X) \ \text{is expected to follow one same distribution if the image denoising model} \ F \ \text{is good. That is because} F(X) \ \text{tries to nearly output the clean image} Y. \ \text{If} X - Y \ \text{follows a distribution,} \ X - F(X) \ \text{should also follow the same one as long as} F(X) \approx Y. \ \text{In Eq. 7, noise maps} \ \{N^{(i)}\}_{i=1}^3 \ \text{are sampled from the same noise model. If the removed noises} \ \{\tilde{N}(X^{(i)})\}_{i=1}^3 \ \text{from}
different images are of diverse distributions, no matter what distribution \( \{N^{(i)}\}_{i=1}^{q} \) is drawn from, there would be a gap between \( N \) with \( \tilde{N}(X) \), i.e., \( \mathbb{E}_{X,Y}[||N - \tilde{N}(X)||^2] \) is large. Considering the lower bound of Eq. (7), the ideal case is that the distribution of \( N \) is exactly the MLE of samples in \( \tilde{N}(X) \), i.e., \( \mathbb{E}_{X,Y}[||N - \tilde{N}(X)||^2] = 0 \). Here we adopt it as our estimation.

Formally, given \( q \) noisy images \( X = \{X^{(i)}\}_{i=1}^{q} \), the denoising model \( F \) outputs \( q \) removed noise maps \( \mathbf{n} = \{\tilde{N}(X^{(i)})\}_{i=1}^{q} \), where \( \tilde{N}(X^{(i)}) = X^{(i)} - F(X^{(i)}) \in \mathbb{R}^{m} \). Let \( \mathbf{n} = \{\tilde{N}(X^{(i)})\}_{i,j=1}^{q,m} \) denote all removed noise pixels, where \( \tilde{N}(X^{(i)}) \) is the value of the \( j \)-th pixel in \( \tilde{N}(X^{(i)}) \). We derive the MLE of samples in \( \mathbf{n} \), shown in the lemma below, where the proof is given in Appx. B.

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

\[
P(n^{*} = \tilde{N}^{(i)} = (mq)^{-1} \forall \tilde{N}^{(i)} \in \mathbf{n},
\]

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

In statistics, to parameterize a given form of distribution, the output of the MLE makes the given samples most probable [24]. For Eq. (7), by replacing the real noise map \( N \) from unknown distribution \( \mathcal{N} \) with MLE \( \mathcal{N}^{*} \) based on \( \tilde{N}(X) \), we get a smaller but tractable estimation of \( \mathbb{E}_{N,X}[||N - \tilde{N}(X)||^2] \):

\[
\mathbb{E}_{N^{*},N}[||N^{*} - \tilde{N}(X)||^2] = \mathbb{E}_{N^{*}}[\mathbb{E}_{X}[\sum_{j=1}^{m}(N^{*}_{j} - \tilde{N}(X)_{j})^2]].
\]

The MLE enables us to estimate the expectation using Monte Carlo integration. Note that each sampled noise map \( N^{*} \) has \( m \) pixels, which could have arbitrary indices. We mathematically prove that the lower bound of Eq. (9) is the noise map \( N^{*} \) sorted in ascending order. The lemma is shown as follows and the proof is in Appx. C.

**Lemma 3.** Given the sampled noise map \( N^{*} \) from \( \mathcal{N}^{*} \), we sort the \( m \) pixels of the removed noise map \( \tilde{N}(X) \) \( (\{\tilde{N}(X)\}_{j=1}^{m} \) in increasing order and define the index list as \( \{v_{1}, v_{2}, \ldots, v_{m}\} \), i.e., \( \tilde{N}(X)_{v_{1}} \leq \tilde{N}(X)_{v_{2}} \leq \ldots \leq \tilde{N}(X)_{v_{m}} \). Similarly, we define the index list for increasingly sorted sampled noise pixels \( \{N^{*}_{j}\}_{j=1}^{m} \) as \( \{v_{1}, v_{2}, \ldots, v_{m}\} \). We have:

\[
\mathbb{E}_{N^{*}}[\mathbb{E}_{X}[\sum_{j=1}^{m}(N^{*}_{j} - \tilde{N}(X)_{j})^2]] \\
\geq \mathbb{E}_{N^{*}}[\mathbb{E}_{X}[\sum_{j=1}^{m}(N^{*}_{v_{j}} - \tilde{N}(X)_{v_{j}})^2]].
\]

Note that Eq. (10) can be unbiasedly estimated by Monte-Carlo (MC) integration with samples from \( \mathcal{N}^{*} \).

In summary, Eq. (10) provides an estimation of lower bound on \( \mathbb{E}_{X,Y}[||F(X) - Y||^2] \) with Eq. (9) as the stepping stone. Then, we finally get a tractable estimation on the upper bound of \( \sigma_{n} \) after applying the lower bound on \( \mathbb{E}_{X,Y}[||F(X) - Y||^2] \) into Eq. (6).

### 3.2. Noise2Info Training

#### 3.2.1 The Procedure of Estimating \( \sigma_{n} \)

The upper bound of \( \sigma_{n} \) has been introduced in Sec. 3.1. In Algo. 1, we show the steps of estimating \( \sigma_{n} \) for a fixed model \( F \). Note that the estimation is utilized as \( \sigma_{loss} \) in Eq. (1). Thus, Noise2Info can avoid manually fixing a value to \( \sigma_{loss} \) as Noise2Same does. Instead, Noise2Info uses an estimate close to \( \sigma_{n} \), which may further improve the denoising performance.

For each noisy image \( X \), we derive its removed noise map \( \tilde{N}(X) = F(X) - X \), of which pixels are further sorted and collected to array \( \mathbf{n} \). In each round of Monte Carlo integration, uniform sampling of \( \mathbf{n} \) is exactly a sample of the maximum likelihood estimation derived in lemma 2. When we sort the sample and calculate its \( L_{2} \) norm with regard to \( \mathbf{n} \), we get a sample for the Eq. (10), which is accumulated in \( E_{l} \). \( E_{l} \) is divided by \( k_{mc} \) as an expectation estimation of \( \mathbb{E}_{X,Y}[||F(X) - Y||^2] \). The terms on the right hand side of Eq. (6) are estimated, which outputs an upper bound of \( \sigma_{n} \).

#### 3.2.2 Training Framework

Sec. 3.2.1 introduces how to estimate \( \sigma_{n} \) for a fixed model \( F \). During training, the deep learning method updates the
Algorithm 2 Noise2Info

Input: The denoising model $\mathcal{F}$, noisy images $X = \{X^{(i)}\}_{i=1}^P$, the number of epochs $k_r$, the number of samples for model updation $k_t$, and $\sigma_n$ estimation $k_u$.

Initialize $\sigma_{\text{loss}} \leftarrow 1$.

for $t \leftarrow 1$ to $k_t$, do

Update $\mathcal{F}$ via loss $\mathcal{L}_{\text{in}} + 2\sigma_{\text{loss}} \mathcal{L}_{\text{ex}}$ with $k_t$ samples.

if $\sigma_{\text{loss}}^t < \sigma_{\text{loss}}^*$ then

$\sigma_{\text{loss}} \leftarrow \sigma_{\text{loss}}^*$

end if

end for

Return: model $\mathcal{F}$ for denoising

---

Figure 3: Training framework of Noise2Info: (A) update the denoising model $\mathcal{F}$ and (B) update the estimation $\sigma_{\text{loss}}$.

model every batch, where we may obtain many $\mathcal{F}$s. Among all estimations on the upper bound of $\sigma_n$, we prefer to choose models that can implicitly estimate tighter bounds (i.e., $\sigma_{\text{loss}} - \sigma_n$ is small). In the following proposition, we state the relationship between $\mathcal{F}$ and its estimation $\sigma_{\text{loss}}$, where proof is provided in Appx. D.

**Proposition 1.** Assume that training model $\mathcal{F}$ under loss (5) pushes output $\mathcal{F}(X)$ closer to clean image $Y$. A more well-trained $\mathcal{F}$ will estimates a smaller $\sigma_{\text{loss}}$ (i.e., tighter upper bound of $\sigma_n$).

Based on above proposition, we can estimate tighter $\sigma_{\text{loss}}$ when using a well-trained $\mathcal{F}$. Correspondingly, as discussed in Sec. 1, we can train more powerful $\mathcal{F}$ while feeding tighter estimation to Eq. (1). Recall the motivational observation in Fig. 2. Fig. 2 (i) demonstrates that the MLE of noise maps removed by $\mathcal{F}_d(\cdot)$ (only trained with 10 batches) cannot well fit the real one, but that of noise maps removed by well-trained $\mathcal{F}_g(\cdot)$ can fit as shown in Fig. 2 (j). This verifies that the gap between $N$ with $\hat{a}$ is small as long as the denoising model can nearly output the clean images.

Therefore, we propose to alternatively train the model $\mathcal{F}$ and estimate $\sigma_n$. We summarize the whole training framework of Noise2Info in Algo. 2 and Fig. 3. In each round, $\mathcal{F}$ is first updated with an upper bound of loss function (5), where $\sigma_{\text{loss}}$ is applied instead of the unknown $\sigma_n$ (Fig. 3(A)). Then, the model $\mathcal{F}$ is fixed and the gradient descent is turned off. $k_u$ samples of noisy images are used for $\sigma_{\text{loss}}$ updation according to Algo. 1(Fig. 3(B)). As each estimation is an upper bound of $\sigma_n$, we update the $\sigma_{\text{loss}}$ once a smaller estimation is found.

4. Experimental Study

4.1. Experimental Setup

**Dataset.** We adopt the benchmark datasets including ImageNet ILSVRC 2012 Val [25], Hánzi [3], and BSD68[20], which are widely adopted in previous works [29, 3, 17, 8]. Besides that, we also conduct experiments on real world datasets SIDD [2], PolyU [30], and discuss another dataset Planaria in Appx. E. ImageNet are colored natural images injected with noise including Poisson noise ($\lambda = 30$), additive Gaussian noise ($\mu = 0, \sigma = 60$), and Bernoulli noise ($p = 0.2$). Hánzi dataset is generated by adding noise to grey images of Chinese characters, where the main experiment applies Gaussian noise ($\mu = 0, \sigma = 0.7$) and Bernoulli noise ($p = 0.25$). The output images are further clipped into $[0, 1]$ (set values to 1 if they are larger than 1 and to 0 if they are smaller than 0). BSD68 dataset contains grey natural images with only Gaussian noise ($\sigma = 25$). We summarize the statistic of these datasets in Tab. 1.

**Baselines and Implementations.** We compare Noise2Info with traditional methods NLM [4] and BM3D [10], supervised methods Noise2True defined in Noise2Noise [19], self-supervised methods Noise2Void [17], Noise2Self [3], ComBS [18], and Noise2Same [29]. We follow the Noise2Same, Noise2Void, and Noise2Self to use Uniform Pixel Selection as the masking strategy. The real values of masked pixels are invisible to the model, which forces the model to use external information to denoise it ($\mathcal{F}$-invariant). Noise2Same replaces the masked pixels with $i$ local average excluding the center pixel (donut) for BSD68; and ii) Gaussian random value for ImageNet and Hánzi, which get the best performance. We show the results under both donut and random replacements, named Noise2Info-D and Noise2Info-R. One can refer to [29] for the detailed setting of other methods.

All the codes are implemented in Tensorflow [1], which are available in the supplementary materials. The train-
Table 2: The comparison of image denoising on three data sets. The best scores are in bold font and the second-best scores are underlined in self-supervised models. ConvBS [18] does not contain the step that requires the noise model. As $\sigma_{loss}$ for Noise2Same can only be set to 1 with unknown $\sigma_n$, we take $\text{Noise2Same}(\sigma_{loss} = \sigma_n)$ as a special category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Model</th>
<th>Data set</th>
<th>ImageNet</th>
<th>Hánzì</th>
<th>BSD68</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional method</td>
<td>Noisy Images without Denoising</td>
<td>9.70</td>
<td>6.45</td>
<td>20.19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NLM [4]</td>
<td>18.04</td>
<td>8.41</td>
<td>22.73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BM3D [10]</td>
<td>18.74</td>
<td>10.90</td>
<td>28.59</td>
<td></td>
</tr>
<tr>
<td>Clean-supervision</td>
<td>Noise2True</td>
<td>23.43</td>
<td>16.00</td>
<td>29.11</td>
<td></td>
</tr>
<tr>
<td>Self-supervision + noise</td>
<td>Noise2Same-R ($\sigma_{loss} = \sigma_n$) [29]</td>
<td>22.57</td>
<td>14.14</td>
<td>27.77</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noise2Same-D ($\sigma_{loss} = \sigma_n$) [29]</td>
<td>22.58</td>
<td>14.08</td>
<td>28.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ConvBS [18]</td>
<td>20.89</td>
<td>10.70</td>
<td>27.15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noise2Void [17]</td>
<td>21.63</td>
<td>13.84</td>
<td>27.28</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noise2Same-R ($\sigma_{loss} = 1$) [29]</td>
<td>22.49</td>
<td>14.38</td>
<td>27.10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noise2Same-D ($\sigma_{loss} = 1$) [29]</td>
<td>22.57</td>
<td>14.36</td>
<td>27.24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noise2Info-R</td>
<td>22.51</td>
<td>14.43</td>
<td>27.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noise2Info-D</td>
<td>22.60</td>
<td>14.43</td>
<td>27.74</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The information of groundtruth $\sigma_n$ and $\sigma_{loss}$ derived by Noise2Info on three data sets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Noise2Info-R</th>
<th>Noise2Info-D</th>
<th>Training Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma_{loss}$</td>
<td>$\sigma_{loss}$</td>
<td>$\mu_n$</td>
<td>$\sigma_n$</td>
</tr>
<tr>
<td>ImageNet</td>
<td>0.9004</td>
<td>0.9015</td>
<td>-1.3726</td>
<td>0.9832</td>
</tr>
<tr>
<td>Hánzì</td>
<td>0.9596</td>
<td>0.9592</td>
<td>-0.1815</td>
<td>1.2193</td>
</tr>
<tr>
<td>BSD68</td>
<td>0.5357</td>
<td>0.5309</td>
<td>0</td>
<td>0.5043</td>
</tr>
</tbody>
</table>

ing is conducted on one machine with 4 NVIDIA V100 GPUs. We follow the setting of Noise2Same [29] to employ GVTNets [27] as the denoising neural network. We set $k_i = 900$ and $k_u = 100$, and the total number of training steps ($(k_i + k_u) \times n_p$) to be the same as Noise2Same.

Evaluation Results. Peak Signal-to-Noise Ratio (PSNR) is used as the evaluation metrics following previous works [29, 19, 17, 3]. For denoising, PSNR is the log-transformation of the ratio between the square of maximum value of a clean image and its mean-squared error against the noisy image: $\text{PSNR}(F(X), Y) = 10 \cdot \log_{10}(\max(Y)^2 / ||F(X) - Y||^2)$, where the larger PSNR value indicates the smaller $||F(X) - Y||^2$, i.e., better image denoising performance.

4.2. Main Empirical Study

We show the main results in Tab. 2 and demonstrate some visual cases in Appx. F1. Among the traditional methods, NLM has weak performance compared with learning-based methods. As discussed in Sec. 2.1, BM3D relies on the additional information $\sigma_n$ for denoising, which leads to better performance and even beats all the self-supervised methods on BSD68. Notably, the mean of noises in BSD68 is zero, which is suitable for BM3D. But learning-based methods still outperform BM3D on the other two datasets. Overall, Noise2True outperforms all the other methods while Noise2Noise outperforms most of the self-supervised methods. However, it is not realistic to assume that either clean images or two noisy images sampled from one clean image are available in the real world.

Among the self-supervised methods, Noise2Info and Noise2Same outperform the other on ImageNet and Hánzì. On BSD68, only Noise2Self with Donut beat them. However, Noise2Self-Donut falls behind our method on ImageNet and Hánzì dataset, which is not stable. Based on Eq. (5), Noise2Same should achieve the best performance when the standard deviation of noise $\sigma_n$ is known. However, Noise2Info (without knowing $\sigma_n$) even outperforms Noise2Same ($\sigma_{loss} = \sigma_n$) on ImageNet and Hánzì. That is because the noises of these two data sets are not zero-mean, which do not follow the assumption of $f$-invariant models. We can also observe that Noise2Same ($\sigma_{loss} = \sigma_n$) is slightly better than Noise2Info on the zero-mean data BSD68. Overall, Noise2Info achieves good performance.
among self-supervised modes without requiring information about clean images, noise, and noise model.

### 4.3. The estimation of $\sigma_n$ in *Noise2Info*

The key component of *Noise2Info* is to estimate $\sigma_n$ as the $\sigma_{\text{loss}}$ in training (see Sec. 3.1 and Algo. 1). As shown in Tab. 3, we list the final derived $\sigma_{\text{loss}}$ of *Noise2Info* on three benchmark data sets. We can observe that the derived $\sigma_{\text{loss}}$ is very close to the true $\sigma_n$. In BSD68, the derived $\sigma_{\text{loss}}$ is slightly higher than $\sigma_n$, which follows the theory in Eq. (6). However, the derived $\sigma_{\text{loss}}$ is even lower than $\sigma_n$ on other data sets because the assumption of zero-mean does not strictly hold. Specifically, the performances of *Noise2Info* (PSNR = 22.51, 22.60) is pretty close to *Noise2Same* (PSNR = 22.49, 22.63) on ImageNet. The reason might be that $\sigma_{\text{loss}}$ and $\sigma_n$ are pretty close to 1. For Hänzi, hard clipping is applied and the $\sigma_n$ is even larger than 1. The results of *Noise2Info* (PSNR = 14.43, 14.43) are slightly higher than those of self-supervised *Noise2Same* (PSNR = 14.38, 14.36) and much higher than those of *Noise2Same* with $\sigma_{\text{loss}} = \sigma_n$ (PSNR = 14.14, 14.08). As BSD68 follows zero-mean Gaussian noise, *Noise2Info* extracts $\sigma_{\text{loss}}$ (0.5357, 0.5309) when $\sigma_n = 0.5043$.

Besides, Proposition 1 states that *Noise2Info*’s estimation will be closer to $\sigma_n$ as training steps increase. Besides, we plot Fig. 4 to show the change of $\sigma_{\text{loss}}$ derivation during training. Note that we add Gaussian noise with zero mean into clean images in Hänzi data since it does not follow the assumption of $J$-invariant models. We can observe that Fig. 4 indeed validates the claims in Proposition 1.

### 4.4. The influence of $\sigma_n$

To further study the influence of $\sigma_n$ to *Noise2Same* with $\sigma_{\text{loss}} = 1$ and *Noise2Info*, we construct noisy images based on the clean images of Hänzi [3], where the added noises are Gaussian noise with different $\sigma_n$. To construct a dataset, each original clean image $Y'$ is applied with a noise map $N'$ with std $\sigma_n'$ to get noisy image $X'$. After normalizing $X'$ to $X$, $\sigma_n'$ is usually different from $\sigma_n$. Assume that the standard deviation of original clean and noisy images are $\sigma_Y$, and $\sigma_{X'}$, we have: $\sigma_n = \sigma_n' / \sigma_{X'} = \sigma_n' / \sqrt{(\sigma_n')^2 + \sigma_{X'}^2}$, where $\sigma_{X'} = (\sigma_n')^2 + \sigma_{X'}$, as noise is independent to the clean images. The model replaces masking values with random samples, and other implementations are the same as Sec. 4.1. As shown in Tab. 4, *Noise2Info* successfully estimates $\sigma_{\text{loss}}$ which closely upper bounds the real $\sigma_n$. When $\sigma_n$ is small, *Noise2Info* stably performs better than *Noise2Same* with $\sigma_{\text{loss}} = 1$ and close to the ideal result from *Noise2Same* with $\sigma_{\text{loss}} = \sigma_n$. With $\sigma_n$ close to 1, the $\sigma_{\text{loss}}$ and PSNR of the 3 methods are close to each other. Except for Gaussian distribution, we further study the influence of more noise distributions in Appx. F2.
4.5. Further Analysis on Limitation of Noise2Info

The theory of Noise2Info mainly follows the assumption of $J$-invariant models where the noise in images is zero-mean and signal independent. Here we further analyze the limitations of Noise2Info caused by such an assumption.

As we discussed in Tab. 3, the upper bound estimation in Noise2Info is not quite accurate in the non-zero mean case. Thus, we use Fig. 3 and Tab. 4 to show that the theory of Noise2Info indeed holds for the zero-mean case. Besides, the empirical performance of Noise2Info is still stable for datasets without the zero-mean assumption. For example, in Tab. 2, Noise2Info is better than other self-supervision methods, though the noise is not zero-mean for Hänzi and not zero-mean nor signal-independent for ImageNet. This validates the generality of Noise2Info.

The signal independent assumption is adopted by many denoising methods [29, 3, 17, 8, 18, 9]. Other than that, some works focus on particular noises such as Poisson-Gaussian noise and Pepper noise to mimic real world scenarios [6, 7]. Therefore, we conduct more experiments on these noises together with non-zero mean assumption in Tab. 5. Poisson-Gaussian noise and pepper noise are both non-zero-mean without fixed variance. For Poisson-Gaussian noise on clean image $Y$, a standard setting is of the form: $X = aP(Y) + N(b)$, where $P(Y)$ is sampled from a Poisson distribution with variance $Y$ and $N$ is sampled from Gaussian distribution $(0, b^2)$. We use 2 groups of parameters $(a, b) = (1, 0.3)$ and $(0.05, 0.02)$, denoted as Poisson-Gaussian A and B, where A is set to be zero-mean. Pepper noise randomly sets a pixel to 0 with $P = 0.25$. As shown in Tab. 5 with the state-of-art work for Poisson-Gaussian noise (FBI [6]), Noise2Info still performs best.

Besides, we conduct experiments on real world datasets, which is discussed in Appx. E. Though inferior to Noise2Same due to special distribution given in Table. 7, our method outperforms other self-supervised methods.

5. Conclusion

In this paper, we propose Noise2Info for self-supervised image denoising, which extracts information of noise only based on noisy images. Compared with methods that require clean images or noisy image pairs, self-supervised models are mostly developed based on the theory of $J$-invariance. It could be further improved if $\sigma_n$ of noise is known. However, it is intractable without the clean images and distribution of noise. In Noise2Info, we first present a theoretical upper bound for $\sigma_n$, and propose a tractable estimation $\hat{\sigma}_n$ only based on noisy images. Then, we prove that the estimation is more accurate when the model is more powerful and propose a training framework that turns this estimation to estimate $\sigma_n$ and update the model. Extensive experiments are conducted on benchmark datasets and synthetic datasets with different scales of noise and different types of noise. The results show that Noise2Info effectively denoises images and tightly bounds the $\sigma_n$.

Acknowledgement

Lei Chen’s work is partially supported by National Key Research and Development Program of China Grant No. 2022YFA0200500 and 2018AAA0101100, the Hong Kong RGC GRF Project 16213620, CRF Project C6030-18G, C1031-18G, C5026-18G, CRF C2004-21GF, AOE Project AoE/E603/18, RIF Project R6020-19, Theme-based project TRS T41-603/20R, China NSFC No. 61729201, Guangdong Basic and Applied Basic Research Foundation 2019B151530001, Hong Kong ITC ITF grants ITS/044/18FX and ITS/470/18FX, Microsoft Research Asia Collaborative Research Grant, HKUST-NAVER/LINE AI Lab, Didi-HKUST joint research lab, HKUST-Webank joint research lab grants and HKUST Global Strategic Partnership Fund (2021 SJTU-HKUST). Shimin Di’s work is supported by the JC STEM Lab of Data Science Foundations funded by The Hong Kong Jockey Club Charities Trust. Shimin Di is the Corresponding Author.
References


