FBI-Denoiser: Fast Blind Image Denoiser for Poisson-Gaussian Noise

Jaeseok Byun*, Sungmin Cha†, and Taesup Moon‡

1Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea
2Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea

{wotjr3868, csm9593}@skku.edu, tsmoon@snu.ac.kr

Abstract

We consider the challenging blind denoising problem for Poisson-Gaussian noise, in which no additional information about clean images or noise level parameters is available. Particularly, when only “single” noisy images are available for training a denoiser, the denoising performance of existing methods was not satisfactory. Recently, the blind pixelwise affine image denoiser (BP-AIDE) was proposed and significantly improved the performance in the above setting, to the extent that it is competitive with denoisers which utilized additional information. However, BP-AIDE seriously suffered from slow inference time due to the inefficiency of noise level estimation procedure and that of the blind-spot network (BSN) architecture it used. To that end, we propose Fast Blind Image Denoiser (FBI-Denoiser) for Poisson-Gaussian noise, which consists of two neural network models: 1) PGE-Net that estimates Poisson-Gaussian noise parameters 2000 times faster than the conventional methods and 2) FBI-Net that realizes a much more efficient BSN for pixelwise affine denoiser in terms of the number of parameters and inference speed. Consequently, we show that our FBI-Denoiser blindly trained solely based on single noisy images can achieve the state-of-the-art performance on several real-world noisy image benchmark datasets with much faster inference time (×10), compared to BP-AIDE.

1. Introduction

Convolutional neural network (CNN)-based denoisers achieved impressive state-of-the-art denoising performances mainly by utilizing the supervised learning approach based on collecting many clean and noisy image pairs. The performance gain was first shown in the additive white Gaussian noise setting [47, 42, 48, 26, 36, 14], then the approach was extended also to the Poisson-Gaussian noise setting, which better models the real-world source-dependent noise. It was demonstrated that the gain was attained not only with respect to the quantitative metrics, e.g., PSNR or SSIM [44], on the real-world noise benchmarks such as DND [35], SIDD [3] and FMD [49], but also with respect to the inference time for denoising (using GPUs), compared to the more conventional prior- or optimization-based methods [17, 21].

Despite above promising achievements, the plain supervised learning approach has a critical drawback in a more practical, real-world setting, since the availability of the enough number of the clean-noisy images pairs for training is sometimes a luxury that cannot be simply assumed. For example, in medical imaging (CT or MRI), obtaining the underlying clean image for a noisy image becomes extremely time-consuming and expensive. In order to overcome this drawback, several attempts have been made in recent years. The first approach is to utilize unpaired clean images and generate synthetic noisy images to again carry on the supervised training with the generated pairs. For example, in [7, 46], based on in-camera signal processing (ISP) pipeline and specific Poisson-Gaussian noise parameters, they generated synthetic noisy sRGB or rawRGB images from clean sRGB images. Another examples can be found in [45, 16], in which they learned a model to generate noise present in the given noisy images, then used that model to corrupt the clean images to build paired supervised training set. While these approaches were shown to achieve good performance for some specific settings, they either lack generalities or have limited performance for real-world noisy image denoising. The second recent approach to remove the requirement of clean-noisy pairs is to train a denoiser solely based on noisy images [25, 23, 6, 24, 38, 34, 13, 41]. However, those methods also had their own limitations, such as requiring pairs of independently realized noisy images for the same clean source [25, 50], poor performance on benchmark datasets [23, 6], large inference time due to requiring many number of samplings [38], or limited or no experiment on real-world noise setting [24, 34, 41, 50, 13].

Recently, BP-AIDE [10], which extends the framework of [12, 11], was proposed as another attempt to lift the requirement of clean images. Namely, the scheme made a
unique combination of Generalized Anscombe Transformation (GAT) [5], Poisson-Gaussian noise estimation [28, 20], and an unbiased estimator of MSE for pixelwise affine denoisers proposed in [11] in order to train a blind denoiser for Poisson-Gaussian noise solely on single noisy images with no additional information. The method achieved the state-of-the-art performance on several benchmarks for real-world noise [49, 3, 35] compared to the methods [23, 17] that operate in the identical setting. However, BP-AIDE had one critical limitation; it suffers from the slow inference time due to the following two reasons. First, for every given noisy image, BP-AIDE must separately carry out the Poisson-Gaussian noise parameter estimation, which usually takes a couple of seconds for moderately sized images.

Second, the method simply utilizes the so-called blind-spot network (BSN) architecture proposed in [12] as a denoiser, but the corresponding structure is quite complex and requires large GPU memory, leading to a slow inference time.

To tackle above limitation, we make two significant improvements on BP-AIDE and propose Fast Blind Image Denoiser (FBI-Denoiser). Firstly, we propose PGE (Poisson-Gaussian Estimation)-Net, which learns the Poisson-Gaussian noise parameters solely from noisy images by converting GAT and Gaussian noise estimation steps to tensor operations and by proposing a novel loss function. Secondly, we devise FBI-Net, a new compact fully convolutional BSN, which performs almost the same as the network in [12] but significantly reduces the inference time. The FBI-Denoiser is trained in two steps; first train PGE-Net with noisy images, then train FBI-Net again with the same noisy images and the outputs of PGE-Net, following the procedures of BP-AIDE. As a result, we significantly improve the inference time of BP-AIDE (×10 speed-up) as well as achieve the state-of-the-art blind denoising performance on real-world noise benchmarks [49, 3, 35].

2. Related Works

Neural network based blind image denoising As mentioned above, several blind image denoisers are proposed to resolve the issue of the dependence on clean images for training. Table 1 summarizes and compares the settings of recently proposed schemes. A variety of schemes [24, 41, 33, 34, 13, 50] were proposed, but their applicability to the Poisson-Gaussian noise was limited. Noise2Noise (N2N) [25] has been shown to be effective for Poisson-Gaussian noise setting, but it still required two independent realizations of noisy images for the same source, which is not practical. For address such limitation of N2N, Noise2void (N2V) [23], Noise2self (N2S) [6], and BP-AIDE [10] adopted self-supervised learning approach which can be solely trained with single images corrupted by Poisson-Gaussian noise. Their settings fully coincide with ours, but they suffer from either poor performance or slow inference time. More recently, D-BSN [45] was proposed for the setting with unpaired clean and noisy images. It also contains self-supervised (self-sup) learning step that improved N2V by elaborating the blind spot network architecture and pixelwise noise level estimation network, but we show that our FBI-Denoiser significantly outperforms it.

<table>
<thead>
<tr>
<th>Alg Requirements</th>
<th>Noisy pairs</th>
<th>Poisson-Gaussian noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ-SSL [23], SURE [11, 33], Noise2Noise [34], GAN2GAN [13]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ext. SURE [36]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>N2N [25]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>N2V [23], N2S [6], BP-AIDE [10], D-BSN [45] (Self Sup), Self2Self [18]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FBI-Denoiser</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Traditional denoising method The classical denoising methods, e.g., wavelet-based [18], filtering-based [8, 17], optimization-based [19, 29, 21] and effective prior-based [51], are typically capable of denoising with only single noisy images. However, since the training procedure with multiple images is absent in these methods, they suffer from large inference time and limited performance.

Noise estimation method Most of above methods assume that prior knowledge about noise characteristics is given, but it is typically unavailable in practice. To alleviate this unrealistic assumption, several noise estimation methods have been proposed, especially for two well-known noise models: additive white Gaussian noise (AWGN) and Poisson-Gaussian noise model. For AWGN, the noise variance of an image is assumed to be constant over all pixel values, i.e., the only parameter is the noise variance. Recently, low-rank patch selection methods [27, 37] using principal component analysis (PCA) showed the state-of-the-art performance in the AWGN case. [15] further refined this approach by resolving the underestimation problem of [27, 37] through statistical analysis of the eigenvalues. Unlike the case of AWGN, the Poisson-Gaussian noise model [20], which is often used to characterize the real source-dependent noise in the raw-sensed images, has a heterogeneous noise variance and two parameters (α, σ). Most existing methods [20, 4, 43, 28] for estimating Poisson-Gaussian noise first obtain the local estimated means and variances, then fit the noise model with these local estimates using maximum likelihood estimation (MLE). [20] first proposed the Poisson-Gaussian noise model and an estimation algorithm for it using wavelet decomposition. Recently, [28] extended this approach to the generalized source-dependent noise by suggesting iterative patch selection method.

3. Problem Setting and Preliminaries

3.1. Notations

We denote \( x \in \mathbb{R}^n \) as the clean image and \( y \in \mathbb{R}^n \) as its noise-corrupted observation. The real-world image sensor
noise is generally modeled with the Poisson-Gaussian noise model \[20\] which consists of two mutually independent components. Under this model, the \(i\)-th noisy pixel becomes

\[ Y_i = \alpha P_i + N_i, \quad i = 1, \ldots, n \] (1)

in which \(P_i \sim \text{Poisson}(x_i)\) is the source-dependent Poisson noise with mean \(x_i\) caused by photon sensing, \(N_i \sim \mathcal{N}(0, \sigma^2)\) is the remaining signal-independent Gaussian noise. Here, \(\alpha > 0\) is a scaling factor which depends on the sensor and analog gain, and \(\sigma\) is the standard deviation of the Gaussian noise. For each clean and noisy image, we assume that each pixel is normalized and clipped to have a value in the range \([0, 1]\). Thus, the Poisson-Gaussian noise model is characterized by the parameters \((\alpha, \sigma)\) and the noise variance of \(Y\) can be represented as

\[ \text{Var}[Y | x] = \alpha^2 x + \sigma^2 \] (2)

### 3.2. Generalized Anscombe Transformation (GAT)

A common approach for denoising the noisy image corrupted with Poisson-Gaussian noise is to apply GAT \[5\], which transforms each pixel \(Y_i > 0\) to

\[ G_{\alpha,\sigma}(Y_i) = \frac{2}{\alpha} \sqrt{\alpha Y_i + \frac{3}{8} \alpha^2 + \sigma^2}. \] (3)

The transform (3) is known to stabilize the noise of each pixel of the transformed image, \(G_{\alpha,\sigma}(Y)\), such that it approximately becomes Gaussian with unit variance. Using this property, we can have a simple denoising scheme for Poisson-Gaussian noise. Namely, a general Gaussian denoiser can be applied to \(G_{\alpha,\sigma}(Y)\) to obtain a denoised version \(D\). Then, by denoting \(D_i\) as the \(i\)-th pixel of \(D\), the estimate of the original clean image \(x\) is obtained by applying the Inverse Anscombe transformation (IAT), which is often approximated by \[30\]

\[ I_{\alpha,\sigma}(D) = \frac{1}{4} D^2 + \frac{1}{4} \sqrt{\frac{3}{2}} D^{-1} - \frac{11}{8} D^{-2} + \frac{5}{8} \sqrt{\frac{3}{2}} D^{-3} - \frac{1}{8} - \frac{\sigma^2}{\alpha^2}. \] (4)

Thus, the final denoised image becomes \(\hat{x} = I(D)\). Note both the GAT and IAT require noise parameters \((\alpha, \sigma)\).

### 3.3. BP-AIDE

As briefly mentioned in the Introduction, BP-AIDE \[10\] combined GAT (3) and Poisson-Gaussian noise estimation methods \[28, 20\], which estimates \(\alpha\) and \(\sigma^2\) of (1), with a pixelwise affine denoiser developed in \[12\]. The method consists of 3 steps, (1) Est+GAT (2) Training (3) Inference, and we briefly review each step below.

1. **Est+GAT** Use the Poisson-Gaussian noise estimation methods \[28, 20\] to obtain the estimated noise parameters \((\hat{\alpha}, \hat{\sigma})\). Then, apply GAT (3) with the estimated parameters and obtain the transformed image \(G_{\hat{\alpha},\hat{\sigma}}(Y)\). Then, the normalized version of the transformed image is obtained by \(Z \triangleq \frac{G_{\hat{\alpha},\hat{\sigma}}(Y) - m}{\beta}, \) in which \(m = \min_i G_{\hat{\alpha},\hat{\sigma}}(Y_i)\) and \(\beta = \max_i G_{\hat{\alpha},\hat{\sigma}}(Y_i) - \min_i G_{\hat{\alpha},\hat{\sigma}}(Y_i)\). Note each pixel in \(Z\) has a value in \([0, 1]\) and the variance of the noise becomes approximately \(\beta^{-2}\).

2. **Training** Given the normalized, transformed noisy image \(Z\), BP-AIDE trains a pixelwise affine denoiser \(f(Z; w) \in \mathbb{R}^n\), proposed in \[12\], of which \(i\)-th reconstruction is

\[ f_i(Z; w) = a_1(w, C_{k \times k}^{-1}) \cdot Z_i + a_0(w, C_{k \times k}^{-1}). \] (5)

In (5), \(C_{k \times k}^{-1}\) denotes the \(k \times k\) noisy patch surrounding \(Z_i\) that excludes \(Z_i\), and \(\{a_m(w, C_{k \times k}^{-1})\}_{m=0}^{1}\) are the outputs of a specially designed fully convolutional network with parameter \(w\) that takes \(Z\) as input, but guarantees the exclusion of \(Z_i\) in the output for location \(i\) (i.e., the so-called blind-spot network (BSN)).

Given \(m\) distinct (normalized, transformed) noisy images \(Z = \{Z^{(j)}\}_{j=1}^{m}\), the training of BP-AIDE is done by minimizing

\[ \mathcal{L}(Z; w) = \frac{1}{m} \sum_{j=1}^{m} L_n \left( Z^{(j)}, f(Z^{(j)}; w); (\beta j)^{-2} \right), \] (6)

in which the \(L_n(\cdot; \cdot)\) in (6) is defined in \[11\] as

\[ L_n(Z, f; \sigma^2) \triangleq \frac{1}{n} \sum_{i=1}^{n} \left( \frac{1}{n} \sum_{j=1}^{m} Z^{(j)}_i - f_i(Z; w) \right)^2 + \frac{\sigma^2}{n} \sum_{i=1}^{n} (2a_{1,i} - 1), \] (7)

in which \(f \triangleq f(Z; w)\) and \(a_{1,i} \equiv a_1(w, C_{k \times k}^{-1})\) for notational brevity. Now, it is shown in \[12: \text{Lemma 1}\] that if the noise that generates \(Z\) is additive, independent, zero-mean with homogeneous variance, \(\sigma\), AWGN, then (7) becomes an unbiased estimate of the mean-squared error (MSE) of \(f(Z; w)\) for estimating the underlying clean image. An important point to emphasize is that such unbiasedness only holds for the pixelwise affine denoisers of the form (5) with \(a_m(w, C_{k \times k}^{-1})\) being conditionally independent of \(Z_i\) given \(Z^{-1}\) (i.e., the outputs of a BSN). Since the GAT transformed and normalized image \(Z^{(j)}\) approximately has AWGN with variance \((\beta j)^{-2}\), minimizing (6), which only depends on the noisy images \(Z\) and no underlying cleans, indeed becomes minimizing the unbiased estimate of MSE.

In \[10\], BP-AIDE simply adopted the BSN architecture proposed in \[12\] and trained the network for obtaining the pixelwise slope and intercept \(\{a_m(w, C_{k \times k}^{-1})\}_{m=0}^{1}\) for all \(i\).

3. **Inference** Once the training is done, when denoising a given test noisy image \(Y_t\) at the inference time, the Est+GAT step is first applied to obtain \(Z_t\), then gets denoised by \(f(Z_t, w_{BP-AIDE})\) in the transformed domain. Then, we apply the reverse of the normalization step in Est+GAT to obtain \(D\) and finally apply the IAT (4) with the estimated noise parameters \((\hat{\alpha}, \hat{\sigma})\) obtained in the Est+GAT step.

5770
3.4. Motivation

While BP-AIDE achieved the state-of-the-art denoising performance for real-world noise benchmarks among the methods that only use single noisy images, we observe its critical bottleneck in inference time lies in the Est+GAT step described above. Namely, the Poisson-Gaussian noise parameters \((\alpha, \sigma)\) need to be estimated for each and every given test image, which typically takes in the order of a few seconds using modern CPUs. Moreover, the BSN from [12], denoted by QED network, uses three different kind of masked convolution filters, which results in large memory usage and unnecessary FLOPs that slows down the inference time. As described in the next section, we dramatically improve the inference time of BP-AIDE \((\times 10)\) by learning to estimate the Poisson-Gaussian noise parameters with a separate network (PGE-Net) and devising a novel BSN (FBI-Net) with much simpler structure.

4. Main Method

As illustrated in Figure 1, our proposed FBI-Denoiser consists of two phases: first train PGE-Net for Poisson-Gaussian noise estimation (Phase 1), then train FBI-Net for blind denoising (Phase 2). In the subsequent sections, we first present the intuition of and the training procedure of PGE-Net (Section 4.1), then present in details about the new BSN architecture (Section 4.2).

4.1. Training PGE-Net: Phase 1

As mentioned in Section 3.2, GAT [5] is known to stabilize the noise such that the noise for each pixel in the transformed image, \(G_{\alpha, \sigma}(Y)\), become approximately independent Gaussian with unit variance. We use this property of GAT to build an intuition for training a network to estimate the Poisson-Gaussian noise parameters, \((\alpha, \sigma)\).

Before describing our method, we introduce a few more notations. First, let \(Y\) denote a noisy image as before that is corrupted by the Poisson-Gaussian noise model \((1)\) with true parameter \((\alpha, \sigma)\). Moreover, define \(B\) as a patch extracted from \(Y\) and \(p_B(x)\) as the PDF of the clean \(x\) over \(B\). Moreover, denote \(G_{\hat{\alpha}, \hat{\sigma}}(Y)\) as the GAT-transformed image with estimated \((\hat{\alpha}, \hat{\sigma})\), and the stabilized noise variance of \(G_{\hat{\alpha}, \hat{\sigma}}(Y)\) is denoted by \(\text{Var}(G_{\hat{\alpha}, \hat{\sigma}}(Y)|x)\). Now, in [31, Proposition 1], it has been shown that, under reasonable assumptions, the following set

\[\mathcal{S}_B \triangleq \left\{ (\hat{\alpha}, \hat{\sigma}) : \int \sqrt{\text{Var}(G_{\hat{\alpha}, \hat{\sigma}}(Y)|x)p_B(x)dx} = 1 \right\}\]

becomes a locally smooth curve around the true noise parameter \((\alpha, \sigma)\). With this proposition, [31] showed that when multiple patches, \(\{B_i\}\)’s, are extracted from an image, then the true \((\alpha, \sigma)\) typically lie in the intersection \(\cap_i \mathcal{S}_{B_i}\).

While the findings of [31] can be applied to estimating true \((\alpha, \sigma)\) for a single image, we extend this intuition for devising a neural network-based estimation method that can learn the estimation function from multiple noisy images. In Figure 2, we show that when the patches \(\{B_i\}\)’s are extracted from multiple noisy images that are corrupted with Poisson-Gaussian noise with the same parameter \((\alpha, \sigma)\), the true \((\alpha, \sigma)\) also tends to lie near the intersection \(\cap_i \mathcal{S}_{B_i}\). This observation suggests that we can use patches from multiple noisy images and may learn a neural network model that can directly estimates the noise parameters \((\hat{\alpha}, \hat{\sigma})\) that make the noise variance of \(G_{\hat{\alpha}, \hat{\sigma}}(Y)\) close to 1. Furthermore, when sufficiently many noisy images with various levels of \((\alpha, \sigma)\) are available as a training set, we may also expect the neural network to generalize well to unseen noise parameters.

Our PGE-Net exactly realizes the intuition made above. We first define a neural network, \(h(\cdot; \theta) : Y \rightarrow \mathbb{R}^2\), that takes the Poisson-Gaussian noise corrupted image \(Y\) and outputs the noise parameter estimates \((\hat{\alpha}, \hat{\sigma})\). Moreover, we denote \(\eta(\cdot) : Z \rightarrow \mathbb{R}\) as the function in [15] that estimates the Gaussian noise variance from an input image \(Z\). Then, given \(m\) distinct noisy images \(Y = \{Y^{(j)}\}_{j=1}^{m}\), the loss function for our PGE-Net becomes

\[L_{\text{PGE}}(Y; \theta) \triangleq \sum_{j=1}^{m} \left( \eta(G_{\hat{\alpha}(\theta), \hat{\sigma}(\theta)}(Y^{(j)})) - 1 \right)^2,\]

in which \(\hat{\alpha}(\theta) \triangleq h_1(Y^{(j)}; \theta)\) and \(\hat{\sigma}(\theta) \triangleq h_2(Y^{(j)}; \theta)\) are the estimated noise parameters \((\hat{\alpha}, \hat{\sigma})\) that are outputs of

Figure 1. Overall process of FBI-Denoiser.
Figure 2. $S_B$ curves for ten patches each cropped from different images randomly selected from the MIT-Adobe FiveK Dataset [9] (True $(\alpha, \sigma) = (0.01, 0.01)$ denoted by x.) $h(\cdot, \theta)$. For the specific architecture of $h(\cdot, \theta)$, we used a three layer U-Net [39] architecture and average pooling layer in the output layer.

Now, when using (8) as a loss function to minimize for training $h(\cdot, \theta)$, an important point to check is whether all the operations for computing the loss is differentiable with respect to the network parameter $\theta$ and can be implemented with tensor operations. To that end, we can see that the GAT (3) can be tensorized by a linear operation and applying an element-wise function. However, implementing the Gaussian noise variance estimation $\eta(\cdot)$ [15] with tensor operations is not trivial since it involves iterative for-loop procedures. To that end, we utilized several tricks using the lower triangular matrix multiplication and masking scheme to replace iterative procedures with tensor operations. Details about the implementations can be found in the Supplementary Material (S.M).

Note the main purpose of PGE-Net is not to simply estimate the noise parameters in a myopic way, but to estimate them fast such that the GAT-transformed image (with the estimated parameters) has stabilized noise variance close to 1 for the next denoising step. Once our PGE-Net is learned via minimizing (8), it enjoys extremely fast inference time (using GPU) as only simple forward pass of the neural network is required to estimate the noise parameters $(\hat{\alpha}, \hat{\sigma})$. Moreover, as we show in Section 5.2, our method becomes much more stable than the conventional noise estimation methods since PGE-Net can smoothly generalize from multiple noisy images, whereas the methods in [20, 28], which run optimization routines separately for each image, may fail to correctly estimate depending on the input image.

4.2. Training FBI-Net: Phase 2

Once the noise parameters $(\hat{\alpha}, \hat{\sigma})$ are estimated with PGE-Net, our denoising network FBI-Net is trained following the same training procedure of BP-AIDE described in Section 3.3 (as also shown in Phase 2 of Figure 1); namely, minimize (6) with the transformed (with the estimated $(\hat{\alpha}, \hat{\sigma})$) and normalized images $Z^{(j)}$. Recall that the BSN architecture was necessary for implementing the pixelwise affine denoiser (5) and maintaining the unbiasedness of (7), and we devise a more efficient BSN in this section.

The ideal condition for BSN is to have a simple structure that excludes only $Z_i$ when computing the activation value at location $i$ in any layer. In practice, some methods implemented BSN by discarding some more pixels other than $Z_i$ when computing the activation value at location $i$, but obviously, the more we exclude, the worse the performance becomes. Recently, several network architecture to enhance the efficiency of BSN were proposed [12, 23, 24, 45] in the denoising context. N2V [23] maintains above constraint indirectly by adopting randomly masking scheme. The masking scheme severely degrades training efficiency since only few pixels can be utilized. HQ-SSL [24] suggests more efficient constrained model by adopting four-way rotations as shown in Figure 3(a). But the requirement of four rotated input images gives limited efficiency. D-BSN [45] devised a completely re-designed model. As shown in Figure 3(b), they mainly use two different types of a masked CNN layer and 1×1 convolution layer to maintain the constraint. This model significantly improves efficiency over previous methods, however, it still has the disadvantage of having a lot of unwanted holes excluded from the receptive field. As shown in Figure 3(c), FC-AIDE [12] designed three different layers, denoted as Q, E and D layer. By utilizing three layers, they obtain the ideal condition we described above. However, since each of the three layers has to be stacked separately for getting it, the model is relatively complex and its inference time is slow.

We propose a more efficient BSN architecture (FBI-Net) by re-designing three different zero-masked convolution layers, $L^1$, $L^2$ and $L^3$ as described in Figure 3(d). $L^1$ is a center masked $3 \times 3$ size of convolution layer which is equal with the first layer proposed in D-BSN. $L^2$ is $5 \times 5$ size of convolution layer which is all masked except for eight holes and $L^3$ is $7 \times 7$ size of convolution layer which only has weights at the center and each edge. Note that $L^3$ can be replaced with $3 \times 3$ size of a dilated convolution layer (dilation = 3). By stacking these layer sequentially, we can get an almost full receptive field with fewer holes than D-BSN. Compared with FC-AIDE, even though FBI-Net have some holes in the receptive field, our layers have the advantage of obtaining a simpler architecture and fewer parameters. Figure 3 shows Table 2. The comparison of BSN on an image of size 512×512.
we used PReLU [22] for all activation functions and Residual Module (RM) proposed in Figure 3(f). To maintain the constraint, we only used $1 \times 1$ CNN layer for the output layer and RM. For all experiments, we set $L = 17$ which has a receptive field of $181 \times 181$ size. Note that we add two different residual connections denoted by Inner and Outer residual connection. We found out that these residual connections make more stable training and improve the denoising performance as can be shown in the ablation study. Table 2 shows the performance comparison of each BSN when denoising an image of size $512 \times 512$. Compared to the baselines, the proposed FBI-Net uses much smaller number of parameters and achieves the fastest inference time while the GPU memory requirement is small.

5. Experimental Results

5.1. Data and Experimental settings

**Data and Implementation details** In synthetic noise experiments, we used two datasets: BSD400 [32] for grayscale images and MIT-Adobe FiveK Dataset (Fivek) [9] for raw-RGB images. For evaluation of the grayscale images, the standard BSD68 dataset [40] was used. For Fivek, which is composed of 5,000 images, 4,800 images were used for training, and the remaining 200 were used as a test set. To reflect the real noise characteristics, various levels of Poisson-Gaussian noise were simulated. Note each model is individually trained and tested with designated noise characteristics. For real noise experiments, we used three real-world noisy image datasets: Fluorescence Microscopy Denoising (FMD) dataset [49], which is composed of grayscale microscopy images, and SIDD [3] / DND [35] which consist of raw-RGB and sRGB images. In FMD, raw noisy images from three separate general configurations are used; confocal FISH (CF FISH), confocal MICE (CF MICE) and two-photon MICE (TP MICE). The training on SIDD and DND was done using only raw-RGB images, and we received the evaluation results on both raw-RGB and sRGB test sets by submitting the results to the public websites [2, 1], respectively. Note that SIDD provides training, validation and test datasets, but DND only provides test images to the public. For training a model, we generated $m = 25,000$ noisy patches of size $210 \times 210$ from each dataset. To reflect the complexity of real noise, we reduced the range of slope coefficient $a_1(\cdot)$ in (5), i.e., we changed from $[0, 1]$ which is the original range of $a_1(\cdot)$ suggested by [10] to $[0, 0.1]$. For a fair comparison, we used the same range $[0, 0.1]$ for both BP-AIDE and FBI-Denoiser in all experiments. The details on the software platform, training and hyperparameters are in the S.M.

**Baselines** We first compared the accuracy of the noise parameters from PGE-Net (phase 1) with two representative baselines: Foi [20] and Liu [28]. Then, we compared the overall denoising performance of FBI-Denoiser with following baselines: GAT+BM3D [17], N2V [23], D-BSN [45], and BP-AIDE [10]. GAT+BM3D is a traditional method, but it is still a very powerful baseline for the Poisson-Gaussian noise denoising. The estimation method in Liu [28] is used as a noise estimation method for GAT+BM3D and BP-AIDE, since it achieves robust performance in real-world noise benchmarks. Moreover, the self-supervised step of D-BSN is used as a baseline for fair comparison. As an upper bound, we trained a model in a supervised way (i.e., using clean target images), denoted as "Sup", that uses the same network architecture as FBI-Net. All results of the baselines and FBI-Denoiser are shown in Figure 4. The layers of previously proposed blind spot network and our FBI-Net.
are reproduced with publicly available source codes, and for
brevity, FBI-Denoiser is denoted as “FBI-D” in this section.

5.2. Experimental results on noise estimation

Here, we first validate the effectiveness of the noise esti-
mation of PGE-Net. Table 3 shows the average of the esti-
imated $(\hat{\alpha}, \hat{\sigma})$ values, obtained by Foi [20], Liu [28] and
PGE-Net from BSD68 images that are corrupted by four dif-
ferent noise levels specified by each row in the table. In

Table 3. The average values of estimated $\hat{\alpha}$ and $\hat{\sigma}$ for BSD68. Bold
denotes the best result among the three methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0.01,0.02)$</td>
<td>$0.042$</td>
<td>$0.001$</td>
<td>$0.042$</td>
</tr>
<tr>
<td>$(0.05,0.02)$</td>
<td>$0.042$</td>
<td>$0.001$</td>
<td>$0.080$</td>
</tr>
<tr>
<td>$(0.05,0.02)$</td>
<td>$0.042$</td>
<td>$0.001$</td>
<td>$0.080$</td>
</tr>
</tbody>
</table>

addition, we indirectly measured the performance of noise esti-
mation by comparing the denoising performance (on
BSD68 and Fivek) of GA T+BM3D that uses the estimated
noise parameters from each estimation method, as shown in
Table 4. Moreover, the performance of GA T+BM3D us-
ing the ground truth noise level is reported in the table as
an upper bound. Firstly, from Table 3, we observed that
unlike for $\alpha$, PGE-Net seems to significantly underestimate
$\sigma$. However, we observe from Table 4 that GA T+BM3D
with the estimated parameters of PGE-Net still shows com-
petitive denoising performance compared to others [20, 28]
with much faster inference time (×2000). This suggests that
the underestimated $\hat{\sigma}$ of PGE-Net has little impact on the
denoising performance of GA T+BM3D or BP-AIDE frame-
work (which carry out GAT using $\hat{\sigma}$). As we emphasized

Table 4. Denoising results of GA T+BM3D with the estimated noise
parameters. Red and blue denote the highest and the second highest
result among the noise estimation algorithms, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
<th>GA T+BM3D</th>
<th>BP-AIDE</th>
<th>D-BSN</th>
<th>FMD (Real)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSD68 $(\alpha,\sigma)$</td>
<td>44.31</td>
<td>0.979</td>
<td>44.61</td>
<td>0.981</td>
<td>44.43</td>
<td>44.80</td>
</tr>
<tr>
<td>Fivek $(\alpha,\sigma)$</td>
<td>46.43</td>
<td>0.921</td>
<td>46.63</td>
<td>0.924</td>
<td>46.80</td>
<td>47.00</td>
</tr>
</tbody>
</table>

the standard deviations of $\hat{\alpha}$’s obtained by PGE-Net are far
smaller than those of others, and, in particular, the failure
case (in which $\hat{\alpha}$ goes to zero) does not occur in PGE-Net.
Note such failure case of $\hat{\alpha}$ is fatal for training BP-AIDE or
FBI-Denoiser since the extremely small value of $\hat{\alpha}$, which is a
denominator of GAT (3), causes the gradient explosion.

5.3. Experimental results on denoising

Synthetic noise The experimental results on synthetic raw-
RGB Fivek dataset are shown in the top of Table 5. We simu-
lated not only the specified levels of $(\alpha, \sigma)$, but also a mixture
of noise levels denoted as “Mixture noise”, which were gener-
ated by following the same procedure as [7] for sampling multiple
noise levels. From the results, we first note that our FBI-D performs very well (often the best) compared to other baselines across various noise levels, including the “Mixture noise” case, with a fast inference time. From this result, we can confirm again that PGE-Net is very effective for FBI-D to achieve a competitive denoising performance. Secondly, BP-AIDE and GA T+BM3D also achieve good results, however, these methods have very slow inference times due to the computational cost of the noise estimation method in Liu [28]. Finally, D-BSN achieves a good performance for the weak noise cases, which surpasses FBI-D, however, when the noise is mixed or strong, D-BSN is far inferior to others. In S.M, we analyze why FBI-D achieves relatively low performance in the weak noise cases.

Table 5. PSNR(dB)/SSIM on Fivek (Synthetic) / FMD (Real) validation
dataset. The colored texts are as before.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
<th>GA T+BM3D</th>
<th>BP-AIDE</th>
<th>D-BSN</th>
<th>FMD (Real)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic $(\alpha,\sigma)$</td>
<td>44.31</td>
<td>0.979</td>
<td>44.61</td>
<td>0.981</td>
<td>44.43</td>
<td>44.80</td>
</tr>
<tr>
<td>Mixture noise</td>
<td>46.43</td>
<td>0.921</td>
<td>46.63</td>
<td>0.924</td>
<td>46.80</td>
<td>47.00</td>
</tr>
<tr>
<td>Real (PSNR/SSIM)</td>
<td>46.10</td>
<td>0.904</td>
<td>46.30</td>
<td>0.906</td>
<td>46.40</td>
<td>0.908</td>
</tr>
</tbody>
</table>

Figure 4. Histogram of estimated $\hat{\alpha}$ for BSD68 from each estima-
tion method.

in Section 4.1, we believe this somewhat counter-intuitive result is possible since the underestimation of PGE-Net for $\sigma$ turns out to be inconsequential for ensuring that the GAT-
transformed image (with the estimated parameters of PGE-
Net) has stabilized noise with homogeneous variance close to
1. To verify this more in detail, we conducted an experiment
using an additional toy example in S.M.

Furthermore, Figure 4 shows the histograms of the $\hat{\alpha}$
values (on BSD68) for two different noise levels obtained by
each estimation method. From the figure, we observe that

Real-world noise The results on FMD dataset are reported in
the bottom of Table 5. From the table, we observe again that
FBI-D consistently outperforms other baselines. More-
over, D-BSN suffers from serious performance degradation, and
this illustrates another example of poor applicability of
D-BSN when the noise is strong.
Table 6 shows the results on SIDD [3] and DND [35] datasets. In DND, since the 50 noisy test images are available, we regard this set as both training and test set. Note this is perfectly possible since all methods in Table 6 do not require any clean images for training. In addition, particularly for the DND results, we also report the results of BP-AIDE and FBI-D, which are trained on the training set of SIDD. For clarity, we present two versions of BP-AIDE and FBI-D trained on different datasets; “(SIDD)” and “(DND)” stand for the dataset used for training, respectively. From the table, firstly, we reconfirm the similar tendency as Table 5; i.e., FBI-D mostly dominates other baselines and enjoys fast inference time. Note SIDD and DND datasets contain a mixture of various noise levels, hence, this result shows the robustness and effectiveness of our FBI-D for the real “mismatched” noise case. Secondly, from the strong performance of “FBI-D (SIDD)” for both SIDD and DND, we verify the strong generalization capability of FBI-D. We also note that, as we show in the S.M., the performance of “FBI-D (SIDD)” is also competitive with supervised trained models. Finally, the visualization results in Figure 5 re-emphasize the strength of “FBI-D (SIDD)”; it reconstructs the detailed texture much better than any other baselines.

Table 6. PSNR/SSIM on SIDD and DND dataset. The colored texts are as before.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SIDD</th>
<th>DND</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAW</td>
<td>GAT+BM3D</td>
<td>N2V</td>
</tr>
<tr>
<td>RAW</td>
<td>42.74</td>
<td>40.37</td>
</tr>
<tr>
<td>RGB</td>
<td>41.78</td>
<td>39.45</td>
</tr>
<tr>
<td>Noisy image</td>
<td>Noisy patch</td>
<td>GAT+BM3D</td>
</tr>
<tr>
<td>Figure 5. Visualization results of DND</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.4. Ablation study

Ablation study on PGE-Net Here, we analyze the effect of the loss function of PGE-Net (8) by comparing with the naive supervised estimation model. Table 7 shows the PSNR/SSIM values of GAT+BM3D using estimated noise parameters from different methods on Fivek dataset corrupted with a mixture of noise levels ($\alpha$ $\in$ $[0, 0.16]^2$, $\gamma$ $\in$ $[0, 0.06]$). For comparison, we trained “Sup (MSE)” with exact same architecture as PGE-Net by minimizing the MSE between ($\alpha$, $\gamma$) and ($\hat{\alpha}$, $\hat{\gamma}$). The result of PGE-Net overwhelming that of “Sup (MSE)” may seem counter-intuitive, but we verify that when GAT is done with the estimated noise parameters of PGE-Net and “Sup (MSE)”, the average variances of the noise in the transformed images become 1.03 and 1.58, respectively. Thus, it turns out that “Sup (MSE)” merely focuses on estimating $\alpha$ and $\gamma$ via minimizing the squared error, but its estimated parameters do not necessarily result in stabilized variance after carrying out GAT using them. We believe this result again shows the effectiveness and strength of our loss function (8) for PGE-Net.

Table 7. Ablation studies of PGE-Net on Fivek (with a mixed noise)

<table>
<thead>
<tr>
<th>Component</th>
<th>FBE</th>
<th>Conv1</th>
<th>Conv2</th>
<th>Conv3</th>
<th>Conv4</th>
<th>Conv5</th>
<th>Conv6</th>
<th>Conv7</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PSNR</td>
<td>43.97</td>
<td>43.11</td>
<td>44.31</td>
<td>44.79</td>
<td>45.24</td>
<td>45.70</td>
<td>45.34</td>
<td>45.34</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9457</td>
<td>0.9519</td>
<td>0.9506</td>
<td>0.9525</td>
<td>0.9564</td>
<td>0.9591</td>
<td>0.9564</td>
<td>0.9564</td>
</tr>
</tbody>
</table>

6. Concluding Remarks

We proposed FBI-Denoser which resolves the computational complexity issue of BP-AIDE by devising PGE-Net, which is much faster than conventional Poisson-Gaussian noise estimation ($\times$2000), and FBI-Net, which is an efficient blind spot network. We showed FBI-Denoser achieves the state-of-the-art blind image denoising performance solely based on “single” noisy images with much faster inference time on various synthetic/real noise benchmark datasets.

Acknowledgment

This work was supported in part by NRF Mid-Career Research Program [NRF-2021R1A2C2007884] and IITP grant [No.2019-0-01396, Development of framework for analyzing, detecting, mitigating of bias in AI model and training data], funded by the Korean government (MSIT).
References


