Photon Limited Non-Blind Deblurring Using Algorithm Unrolling

Yash Sanghvi, *Student Member, IEEE*, Abhiram Gnanasambandam, *Member, IEEE*, and Stanley H. Chan, *Senior Member, IEEE*

***Abstract*—Image deblurring in photon-limited conditions is ubiquitous in a variety of low-light applications such as photography, microscopy and astronomy. However, the presence of the photon shot noise due to the low illumination and/or short exposure makes the deblurring task substantially more challenging than the conventional deblurring problems. In this paper, we present an algorithm unrolling approach for the photon-limited deblurring problem by unrolling a Plug-and-Play algorithm for a fixed number of iterations. By introducing a three-operator splitting formation of the Plug-and-Play framework, we obtain a series of differentiable steps which allows the fixed iteration unrolled network to be trained end-to-end. The proposed algorithm demonstrates significantly better image recovery compared to existing state-of-the-art deblurring approaches. We also present a new photon-limited deblurring dataset for evaluating the performance of algorithms.**

***Index Terms*—Photon limited, poisson deconvolution, deblurring, plug-and-play, algorithm unrolling.**

I. Introduction

IMAGE deblurring is a classical restoration problem where the goal is to recover a clean image from an image corrupted by a blur due to motion, camera shake, or defocus. In the simplest setting assuming a spatially invariant blur, the forward image degradation problem is

$$\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x} + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the clean image to be recovered from the corrupted image $\boldsymbol{y} \in \mathbb{R}^N$, the vector $\boldsymbol{h} \in R^d$ denotes the blur kernel, $\boldsymbol{\eta} \in \mathbb{R}^N$ denotes the additive i.i.d Gaussian noise, and "$*$" denotes the convolution operator. The deblurring problem can be further classified as *non-blind* and *blind*. A non-blind deblurring problem assumes that the blur kernel $\boldsymbol{h}$ is known whereas a blind-deblurring problem do not make such an assumption. In this paper, we focus on the non-blind case.

While non-blind deblurring methods are abundant [1], [2], [3], [4], [5], [6], the majority are designed for well-illuminated scenes where the noise is i.i.d. Gaussian and the noise level is not too high. However, as one pushes the photon level low enough that the photon shot noise dominates, the deblurring task is no longer as simple. As illustrated in Fig. 1, which is a real low-light example we captured using a Canon T6i camera at a photon level approximately 5 lx, the observed image is not only dark but is strongly contaminated by photon shot noise that is visible in the histogram equalized image. To further elaborate on the operating regime of the proposed method, we show in Fig. 2 a comparison between this paper and other mainstream deblurring work. We highlight the raw sensor capture shown in the bottom left of each sub-figure and the tone-mapped image shown in the top right of each sub-figure at different illumination levels.

We refer to the problem of interest as the *photon limited* non-blind deblurring. Photon limited deblurring is a common problem for a variety of applications such as microscopy [7] and astronomy [8]. One should note that photon limited imaging is a problem even if we use a perfect sensor with zero read noise and 100% quantum efficiency. The photon shot noise still exists due to the stochasticity of the photon arrival process [9]. Therefore, the solution presented in this paper is pan-sensor, meaning that it can be applied to the standard CCD and CMOS image sensors and the more advanced quanta image sensors (QIS) [10], [11], [12].

A. Problem Formulation

Consider a monochromatic image $\boldsymbol{x} \in \mathbb{R}^N$ normalized to [0,1]. We write the blurred image as $\boldsymbol{H}\boldsymbol{x}$ where $\boldsymbol{H} \in R^{N \times N}$ represents the blur kernel $\boldsymbol{h}$ in the matrix form. In photon-limited conditions, the observed image is given by

$$\boldsymbol{y} = \text{Poisson}(\alpha \cdot \boldsymbol{H}\boldsymbol{x}), \tag{2}$$

where Poisson$(\cdot)$ denotes the Poisson process, and α is a scalar to be discussed. The likelihood of the observed image $\boldsymbol{y}$ follows the Poisson probability distribution:

$$p(\boldsymbol{y}|\boldsymbol{x};\alpha) = \prod_{j=1}^{N} \frac{[\alpha \boldsymbol{H}\boldsymbol{x}]_j^{[\boldsymbol{y}]_j} e^{-[\alpha \boldsymbol{H}\boldsymbol{x}]_j}}{[\boldsymbol{y}]_j!}, \tag{3}$$

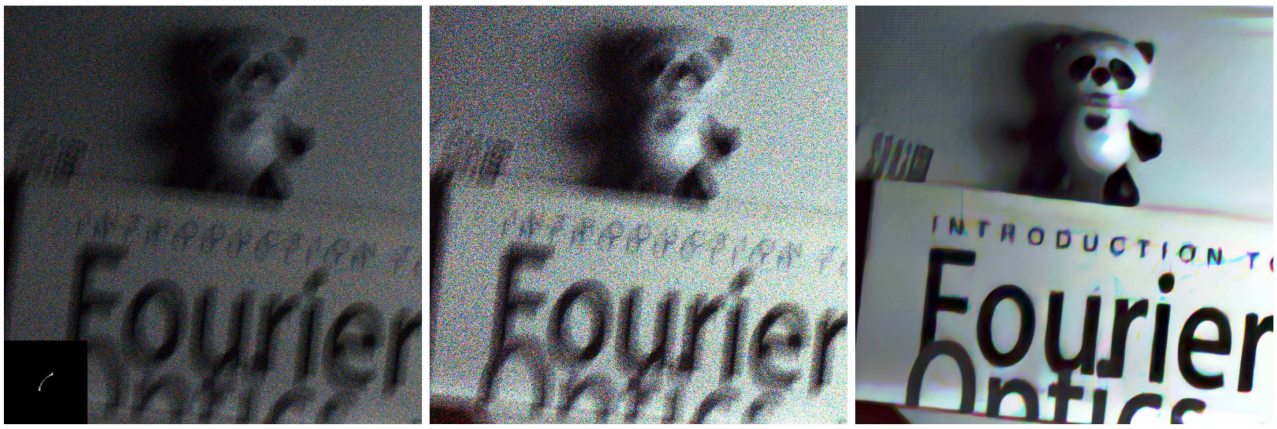
where $[\cdot]_j$ denotes the jth element of a vector. The scalar α represents the photon level. It is a function of the sensor's properties (e.g. quantum efficiency), camera settings (exposure time, aperture), and illumination level of the scene. For a given illumination, the photon level α can be increased by increasing the exposure time or the aperture. To give readers a better idea of

Manuscript received 27 October 2021; revised 10 May 2022 and 21 September 2022; accepted 22 September 2022. Date of publication 26 September 2022; date of current version 5 October 2022. This work was supported by the National Science Foundation under Grants IIS-2133032 and ECCS-2030570. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Petros T. Boufounos. *(Corresponding author: Yash Sanghvi.)*

Yash Sanghvi and Stanley H. Chan are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ysanghvi@purdue.edu; stanchan@purdue.edu,).

Abhiram Gnanasambandam is with the Purdue University, West Lafayette, IN 47907 USA (e-mail: abhiram.g94@gmail.com,).

Digital Object Identifier 10.1109/TCI.2022.3209939



(a) Raw camera image (b) Histogram equalized image (c) Our reconstruction

Fig. 1. Overview. The goal of this paper is to present a new algorithm that reconstructs images from blur at a photon-limited condition.

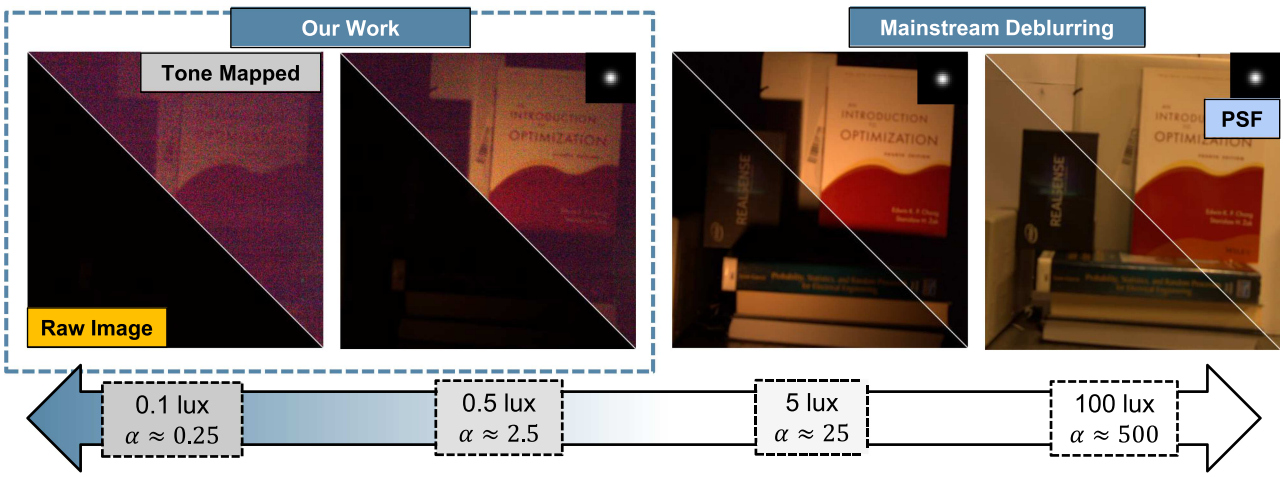


Fig. 2. Comparison of photon-limited scenes (Left) with relatively well illuminated scenes (Right). Raw images and their tone mapped versions taken in different illuminations and blurred by defocus are shown in the figure. As illumination of the scene decreases, the photon shot noise becomes more dominant, making the deblurring problem substantially more difficult - as shown in Fig. 3. In this paper, we address the problem of non-blind deblurring in a *photon-limited* setting i.e. when the number of photons captured by the sensor is low leading to corruption of images by the photon shot noise.

TABLE I
LIGHTING CONDITION AND ILLUMINATION LEVEL

Lighting condition	Illumination (lux)
Sunset	400
Dimly-lit Street	20-50
Moonlight	1
$\alpha = 5$ (This paper)	1

the photon level α, we give a rough estimate of the photon flux (measured in terms of lux level) in Table I under a few typical imaging scenarios.[1]

Under such a severe lighting condition, state-of-the-art algorithms have a hard time working. In Fig. 3 we use the deep Wiener deblurring network [1] to deblur the image. When the illumination is strong, the method performs well. But when the illumination is weak, the algorithm performs poorly. We remark that this observation is common for many mainstream deblurring algorithms.

[1]To estimate the photon level α from the photon flux level, we set the scene illumination to 1 lx (measured using a light meter) and measure the corresponding photons-per-pixel from the image sensor data captured using a Canon EOS Rebel T6i.

B. Contributions and Scope

Photon-limited non-blind deblurring is a special case of the Poisson linear inverse problem. We limit the scope to deblurring so that we can demonstrate the algorithm using real low-light data.

Existing photon-limited deblurring methods are mostly deterministic [13], [14], [15]. To overcome the limitation of these methods, in this paper we present a deep-learning solution. We make two contributions:

1) We propose an unrolled plug-and-play (PnP [16], [17]) algorithm for solving the non-blind deblurring problem in *photon-limited* conditions. Unlike existing work such

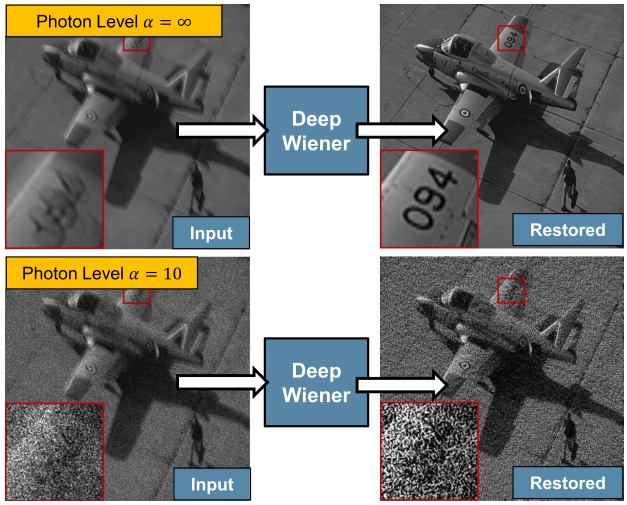


Fig. 3. Limitation of existing image deblurring algorithms when applied to low-light images. In this example we use the pre-trained neural network [1] to recover a well-illuminated scene and a poorly-illuminated scene. The method fails because of the noise, even though the deblurring in a well-illuminated scene is satisfactory.

as [18] which uses an inner optimization to solve the Poisson proximal map, we use a three-operator splitting technique to turn all the sub-routines differentiable. This allows us to train the unrolled network end-to-end (which is previously not possible), and hence makes us the first unrolled network for Poisson deblurring.

2) We overcome the difficulty of collecting *real* photon-limited motion blur kernels and images for algorithm evaluation. A dataset containing 30 low-light images and the corresponding blur kernels are produced. We make this dataset publicly available.

II. Related Work

A. Poisson Deconvolution

Poisson deconvolution has been studied for decades because of its important applications [19]. One of the earliest and the most cited works is perhaps the Richardson-Lucy (RL) algorithm [14], [15]. The method assumes a known blur kernel and derives an iterative scheme which converges to the maximum-likelihood estimate (MLE) of the deconvolution problem. The RL algorithm was applied to problems such as emission tomography [20] and confocal microscopy [21], [22]. However, since the prior is not used, the quality of reconstruction is limited.

Another class of iterative methods is based on maximum-a-posteriori (MAP) estimation by using a signal prior. For example, PIDAL-TV [23] solves a MAP cost function with the total-variation (TV) regularization using an augmented Lagrangian framework. Similarly, the sparse Poisson intensity reconstruction algorithm (SPIRAL) [24] looks for sparse solutions in an orthonormal basis, whereas [25] solves a MAP cost function with multiscale prior using the expectation-maximization algorithm.

Shrinkage based approaches such as PURE-LET [13] assume the deconvolution output to be a linear combination of elementary functions and minimize the expected mean squared error under a joint Poisson-Gaussian noise model. This boils down to solving a linear system of equations and has been also used to solve denoising, deblurring processes under Gaussian noise assumptions [26], [27].

Denoising under Poisson noise conditions can be viewed as a special case of the deblurring problem. One of the widely used techniques for Poisson denoising is the variance stabilizing transforms (VST) which applies the Anscombe transform [28] to stabilize the spatially varying noise variance. A standard denoising method is then used, followed by the inverse Anscombe transform. In [29], it was shown that an optimal inverse transform can outperform other standard Poisson denoising methods such as [30], [31]. The method in [32] provides an iterative version of the denoising via VST scheme by treating last iteration's denoised image as scaled Poisson data.

B. Plug-and-Play

The Plug-and-play (PnP) framework was first introduced in [16] as a general purpose method to solve inverse problems by leveraging an off-the-shelf denoiser. Since then, the framework has been applied to different problems like bright field electron tomography [33] and magnetic resonance imaging (MRI) [34]. Using the same principle but with the half-quadratic splitting scheme, [35] demonstrated the use of a single denoiser for different image restoration tasks such as super-resolution, deblurring, and inpainting. Variations of PnP have also been used for Poisson deblurring [18], [36] and non-linear inverse problems [37]. A stochastic version of the scheme (PnP stochastic proximal gradient method) has been proposed for inverse problems with prohibitively large datasets [38]. Using the consensus equilibrium (CE) framework [39], the scheme can be extended to fuse multiple signal and sensor models.

The convergence of the Plug-and-Play scheme has been studied in detail. For example, [17] provided a variation of the scheme which was provably convergent under the assumptions of a bounded denoiser and its performance was analysed under assumptions of a graph filter denoiser in [40]. [41] showed that if a denoiser satisfies certain Lipshitz conditions, the corresponding Plug-and-Play scheme can be shown to converge. Furthermore, the authors proposed real-spectral normalization as a way to impose the conditions on deep-learning based denoisers.

A closely related method which provides a framework to solve inverse problems using denoisers is REgularization by Denoising (RED) [42], [43]. The framework poses the cost function for an inverse problem as sum of a data term and image-adaptive Laplacian regularization term. This allows the resulting iterative process to be written as a series of denoising steps. In [44], it was mentioned that for RED to be valid the denoiser needs to have a symmetric Hessian.

C. Algorithm Unrolling

The difficulty of running PnP and RED is that they need to iteratively use a deep network denoiser. An alternative way to implement the algorithm was proposed by Gregor and LeCun in 2010 [45] to unroll an iterative algorithm and train it in a

supervised manner. For example, one can unroll the iterative shrinkage threshold algorithm (ISTA) for the purpose of approximating sparse codes of an image. The idea of unrolled networks has been employed in various image restoration tasks such as super-resolution [46], deblurring [47], [48], compressive sensing [49], and haze removal [50]. For a more extensive review of algorithm unrolling, we refer the reader to [51]. More recently, there are new attempts to relax the fixed iteration structure of unrolling by analyzing the equilibrium of the underlying operators [52].

As stated in [51], unrolling iterative algorithms provide multiple advantages compared to generic deep learning architectures. For example, the unrolled networks provide greater interpretability and are often parameter efficient compared to their counterparts such as the U-Net [53]. Since the networks are unrolled version of iterative algorithms, they are less susceptible to problem of overfitting.

III. Method

A. Algorithm Unrolling

The proposed solution for the Poisson deblurring problem is to unroll the iterative PnP algorithm. We start by deriving the PnP steps. In the "unrolled" version of the iterative algorithm, each iteration is treated as a computing block. Each computing block has its own set of trainable parameters. The blocks are concatenated in series with each other. The output at the end of the last block is used as the target for a supervised loss to fine-tune the trainable parameters.

Before describing the iterative algorithm we aim to unroll, we briefly describe the underlying cost function. Most inverse problem algorithm aim to determine the MAP estimate of the underlying signal $\boldsymbol{x}$ by maximizing the log-posterior

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} \left[\log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x})\right], \tag{4}$$

where $p(\boldsymbol{x})$ denotes the natural image prior. Plugging (3) in (4) and taking the negative of the cost function, the maximization becomes

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left[\alpha \mathbb{1}^T \boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}^T \log\left(\alpha \boldsymbol{H}\boldsymbol{x}\right) - \log p(\boldsymbol{x})\right], \tag{5}$$

where $\mathbb{1}$ represents the all-one vector. Note that the factorial term $\log \boldsymbol{y}!$ has been dropped since it is independent of $\boldsymbol{x}$. The prior $p(\boldsymbol{x})$ has not been explicitly specified yet and this issue will be addressed through the use of a denoiser in the next subsection.

B. Conventional PnP for Poisson Inverse Problems

Now we describe how the Plug-and-Play method can be applied to the Poisson deblurring problem. We start with the alternate direction of method of multipliers (ADMM) [54] formulation – where we convert the unconstrained optimization problem to a constrained optimization problem by performing the variable splitting $\boldsymbol{x} = \boldsymbol{z}$

$$\{\boldsymbol{x}^*, \boldsymbol{z}^*\} = \underset{\boldsymbol{x},\boldsymbol{z}}{\operatorname{argmin}} \left[-\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{z})\right], \quad \text{subject to } \boldsymbol{x} = \boldsymbol{z}, \tag{6}$$

At the minimum of the above optimization problem, the constraint $\boldsymbol{x}^* = \boldsymbol{z}^*$ must be satisfied and hence the constrained optimization solution is equivalent to the unconstrained solution in (5).

The augmented Lagrangian associated with the constrained problem in (6) is

$$\{\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*\} = \underset{\boldsymbol{x},\boldsymbol{z}}{\operatorname{argmin}} \Big[\alpha \mathbb{1}^T \boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}^T \log(\alpha \boldsymbol{H}\boldsymbol{x}) - \log p(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|^2 - \frac{\rho}{2}\|\boldsymbol{u}\|^2\Big], \tag{7}$$

where $\boldsymbol{u}$ denotes the scaled Lagrange multiplier corresponding to the constraint $\boldsymbol{x} = \boldsymbol{z}$, and ρ denotes the penalty parameter. The corresponding iterative updates are:

$$\boldsymbol{x}^{k+1} = \underbrace{\underset{\boldsymbol{x}}{\operatorname{argmin}} \Big[\alpha \mathbb{1}^T \boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}^T \log(\alpha \boldsymbol{H}\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}^k\|^2\Big]}_{\text{Proximal operator for the negative log-likelihood}}, \tag{8a}$$

$$\boldsymbol{z}^{k+1} = \underbrace{\underset{\boldsymbol{z}}{\operatorname{argmin}} \Big[-\log p(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{z} - \widetilde{\boldsymbol{z}}^k\|^2\Big]}_{\text{Proximal operator for the negative-log-prior}}, \tag{8b}$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \left(\boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}\right), \tag{8c}$$

with $\widetilde{\boldsymbol{x}}^k \overset{\text{def}}{=} \boldsymbol{z}^k - \boldsymbol{u}^k$ and $\widetilde{\boldsymbol{z}}^k \overset{\text{def}}{=} \boldsymbol{x}^k + \boldsymbol{u}^k$. In the Plug-and-Play framework [16], [17], the $\boldsymbol{z}$ update in (8b) is implemented by an image denoiser.

The difficulty of solving the above problem is that the $\boldsymbol{x}$-update in (8a) does not have a closed form expression for the Poisson likelihood. Thus (8a) needs to be solved using an inner-loop optimization method such as L-BFGS [55]. Unrolling this inner-loop optimization solver can be inefficient as it may not be differentiable. Hence unrolling the PnP scheme for the Poisson inverse problem using the existing framework is infeasible. To be more specific, while the $\boldsymbol{z}$-update in (8b) can be implemented as a neural network and hence is differentiable, the same cannot be said for $\boldsymbol{x}$-update in (8a). As shown in Fig. 4, when (8a) is solved using another iterative method such as L-BFGS (for e.g. in [18]), it is not differentiable. As a result, training the unrolled network via backpropagation is not possible unless (8a) can be made differentiable.

C. Three-Operator Splitting for Poisson PnP

As explained in the previous subsection, the current framework does not allow for algorithm unrolling. To circumvent this issue, we use an alternate three-operator formulation of the PnP-framework. Through this reformulation of Plug-and-Play, we derive a series of iterative updates where each step can be implemented as a single-step that is differentiable. The three-operator splitting strategy we use here has been used in context of Poisson deblurring in [23], [56] and [36] using a TV and BM3D denoiser respectively.

In this scheme, instead of a two-operator splitting strategy for conventional PnP in (6), we use three-operator splitting to form the corresponding constrained optimization problem. Specifically, in addition to splitting the variable as $\boldsymbol{x} = \boldsymbol{z}$, we

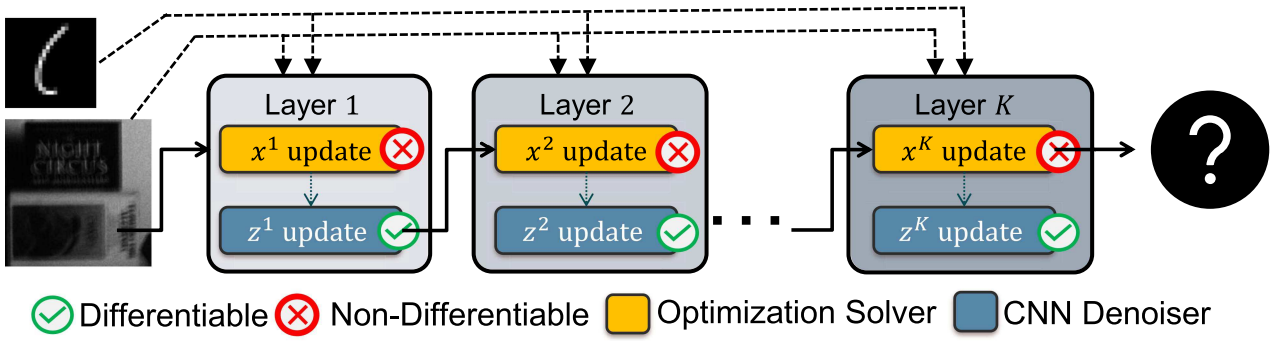


Fig. 4. Conventional two-operator splitting Plug-and-Play. Conventional Plug-and-Play applied to the Poisson deblurring problem using equations (8a) and (8b). While (8b) is implemented as an image denoiser and hence differentiable, $\boldsymbol{x}$-update i.e. (8a) is implemented as a convex optimization solver and hence not differentiable. This makes the conventional PnP infeasible for fixed iteration unrolling and hence end-to-end training.

introduce a third variable $\boldsymbol{v}$ corresponding to blurred image $\boldsymbol{Hx}$ and hence the constraint $\boldsymbol{Hx} = \boldsymbol{v}$.

$$\{\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{v}^*\} = \underset{\boldsymbol{x},\boldsymbol{z},\boldsymbol{v}}{\operatorname{argmin}} \left[-\boldsymbol{y}^T \log(\alpha \boldsymbol{v}) + \alpha \mathbb{1}^T \boldsymbol{v} + \log p(\boldsymbol{z})\right], \quad \text{subject to } \boldsymbol{x} = \boldsymbol{z}, \quad \text{and} \quad \boldsymbol{Hx} = \boldsymbol{v}. \tag{9}$$

After forming the corresponding augmented Lagrangian, we arrive at the following iterative updates:

$$\boldsymbol{x}^{k+1} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left[\frac{\rho_1}{2}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}_0^k\|^2 + \frac{\rho_2}{2}\|\boldsymbol{Hx} - \widetilde{\boldsymbol{x}}_1^k\|^2\right], \tag{10a}$$

$$\boldsymbol{z}^{k+1} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \left[-\log p(\boldsymbol{z}) + \frac{\rho_1}{2}\|\boldsymbol{z} - \widetilde{\boldsymbol{z}}_1^k\|^2\right], \tag{10b}$$

$$\boldsymbol{v}^{k+1} = \underset{\boldsymbol{v}}{\operatorname{argmin}} \left[-\boldsymbol{y}^T \log(\alpha \boldsymbol{v}) + \alpha \mathbb{1}^T \boldsymbol{v} + \frac{\rho_2}{2}\|\boldsymbol{v} - \widetilde{\boldsymbol{v}}^k\|^2\right], \tag{10c}$$

$$\boldsymbol{u}_1^{k+1} = \boldsymbol{u}_1^k + \boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}, \tag{10d}$$

$$\boldsymbol{u}_2^{k+1} = \boldsymbol{u}_2^k + \boldsymbol{Hx}^{k+1} - \boldsymbol{v}^{k+1}, \tag{10e}$$

where $\widetilde{\boldsymbol{x}}_0^k \overset{\text{def}}{=} \boldsymbol{z}^{k+1} - \boldsymbol{u}_1^k$, $\widetilde{\boldsymbol{x}}_1^k \overset{\text{def}}{=} \boldsymbol{v}^{k+1} - \boldsymbol{u}_2^k$, $\boldsymbol{v}^k \overset{\text{def}}{=} \boldsymbol{Hx}^k + \boldsymbol{u}_2^k$, and $\widetilde{\boldsymbol{z}}^k \overset{\text{def}}{=} \boldsymbol{x}^k + \boldsymbol{u}_1^k$. Similar to the PnP formulation described in last subsection, the vectors $\boldsymbol{u}_1, \boldsymbol{u}_2$ denote the scaled Lagrangian multipliers for the constraints $\boldsymbol{x} - \boldsymbol{z} = 0$ and $\boldsymbol{Hx} - \boldsymbol{v} = 0$ respectively. The scalars ρ_1, ρ_2 denote the corresponding penalty parameters.

Each of the subproblems defined in (10a, 10b, 10c) have a closed form solution and are described below:

$\boldsymbol{x}$-subproblem: (10a) is a least squares minimization problem, whose solution can be explicitly given as follows:

$$\boldsymbol{x}^{k+1} = \left(\boldsymbol{I} + (\rho_2/\rho_1)\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\left(\widetilde{\boldsymbol{x}}_0^k + (\rho_2/\rho_1)\,\boldsymbol{H}^T\widetilde{\boldsymbol{x}}_1^k\right). \tag{11}$$

Since $\boldsymbol{H}$ represents a convolutional operator, the operation can be performed without any matrix inversions using Fourier Transforms.

$$\boldsymbol{x}^{k+1} = \mathcal{F}^{-1}\left[\frac{\mathcal{F}(\widetilde{\boldsymbol{x}}_0^k) + (\rho_2/\rho_1)\overline{\mathcal{F}(\boldsymbol{h})}\mathcal{F}(\widetilde{\boldsymbol{x}}_1^k)}{1 + (\rho_2/\rho_1)|\mathcal{F}(\boldsymbol{h})|^2}\right], \tag{12}$$

where $\mathcal{F}(\cdot)$ represents the discrete Fourier transform of the image or blur kernel implemented using the Fast Fourier Transform after appropriate boundary padding. We refer to it as the *deblurring operator*.

$\boldsymbol{z}$-subproblem: (10b) is a proximal operator for the negative log prior term. Using the insight provided in Plug-and-Play scheme, (10b) can be viewed as a denoising operation

$$\boldsymbol{z}^{k+1} = D\left(\widetilde{\boldsymbol{z}}^k\right), \tag{13}$$

where $D(\cdot)$ is any image denoiser. For end-to-end training, we require $D(\cdot)$ to be differentiable and trainable – a property satisfied by all convolutional neural network denoisers.

$\boldsymbol{v}$-subproblem: (10c) is a convex optimization problem but can be solved without an iterative procedure. Separating out each component of the vector minimization and setting the gradient equal to zero gives the following equation

$$-\frac{[\boldsymbol{y}]_i}{[\boldsymbol{v}^{k+1}]_i} + \alpha + \rho_2\left(\left[\boldsymbol{v}^{k+1}\right]_i - \left[\widetilde{\boldsymbol{v}}^k\right]_i\right) = 0, \tag{14}$$

for $i = 1, 2, \ldots, N$. Solving the resulting quadratic equation and ignoring the negative solution gives the following update step

$$\boldsymbol{v}^{k+1} = \frac{\left(\rho_2\widetilde{\boldsymbol{v}}^k - \alpha\right) + \sqrt{\left(\rho_2\widetilde{\boldsymbol{v}}^k - \alpha\right)^2 + 4\rho_2\boldsymbol{y}}}{2\rho_2}, \tag{15}$$

Since the optimization problem in (10c) is a sum of the the negative log-likelihood for Poisson noise and a quadratic penalty term, we refer to this update as *Poisson proximal operator*.

The convergence of Algorithm 1 has been derived in [23]. It was shown that as long as $\boldsymbol{G} = [\boldsymbol{H}^T, \boldsymbol{I}]^T$ has a full column rank, the three-operator splitting scheme converges. Furthermore, assuming the denoiser D is continuously differentiable and $\nabla D(\cdot)$ is symmetric with eigenvalues in [0,1], convergence results in [33] show that the corresponding negative-log prior, i.e., $-\log(p(\cdot))$ is closed, proper and convex. Combined with the result from [23], it can be shown that the three-operator PnP scheme in Algorithm 1 converges.

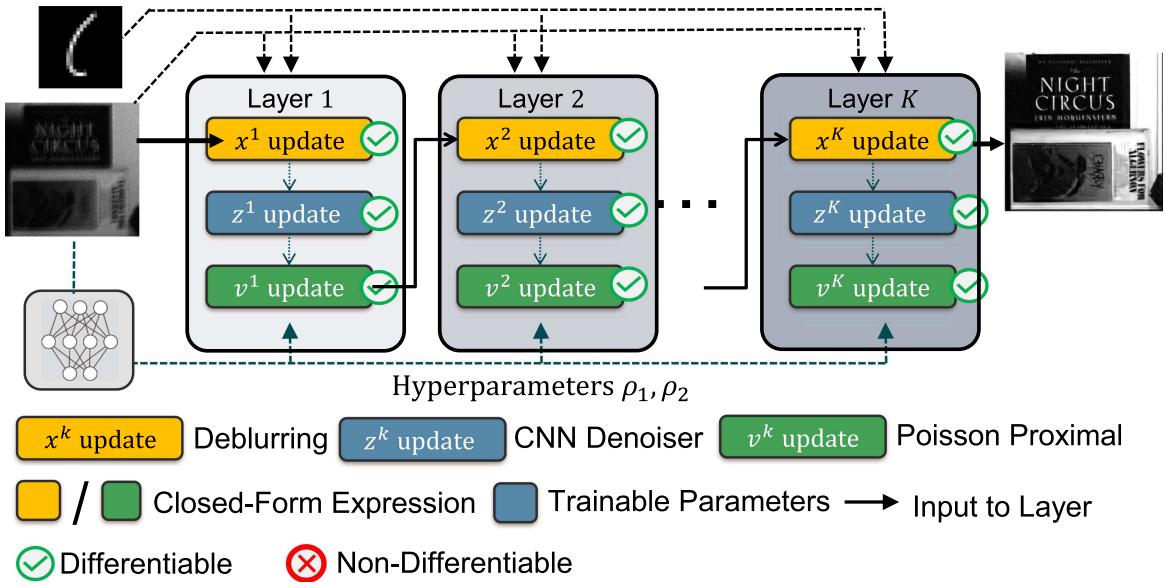


Fig. 5. Proposed unrolled Plug-and-Play for deblurring. For conventional PnP, the data sub-problem cannot be solved in a single step and instead requires convex optimization solvers. This stops us from unrolling the iterative procedure and training it end-to-end via back-propagation. Through the three-operator splitting formulation of the problem, each sub-module in an iteration is in closed form and more importantly, differentiable. This allows for end-to-end training which was not possible in conventional PnP. The network below the input represents the hyperparameter network which predicts ρ_1 and ρ_2 using the blur kernel and the photon level.

Algorithm 1: Three-Operator Splitting for Poisson PnP.

Input: Blurred and Noisy Image $\boldsymbol{y}$, kernel $\boldsymbol{h}$, Photon level α
Initialize $\boldsymbol{x}^0$ using (16)
$\boldsymbol{z}^0 \leftarrow \boldsymbol{x}^0$, $\boldsymbol{v}^0 \leftarrow \boldsymbol{y}$ $\boldsymbol{u}_1^0 \leftarrow 0$, $\boldsymbol{u}_2^0 \leftarrow 0$
for $k = 1, 2, \ldots, K$ **do**
  Update $\boldsymbol{x}^k$ using Eq. (12)
  Update $\boldsymbol{z}^k$ using Eq. (13)
  Update $\boldsymbol{v}^k$ using Eq. (15)
  $\boldsymbol{u}_1^k \leftarrow \boldsymbol{u}_1^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k$
  $\boldsymbol{u}_2^k \leftarrow \boldsymbol{u}_2^{k-1} + \boldsymbol{H}\boldsymbol{x}^k - \boldsymbol{v}^k$
end for
return $\boldsymbol{x}^K$

D. Unfolding the Three-Operator Splitting

With an end-to-end trainable iterative process, we can now describe the unfolded iterative network. The Plug-and-Play updates described in Algorithm 1 are now unfolded for $K = 8$ iterations and the entire differentiable pipeline is trained in a supervised manner, as summarized in Fig. 5. We refer the resulting neural network architecture as **Photon-Limited Deblurring Network** (*PhD-Net*).

Initialization: To initialize the variable $\boldsymbol{x}^0$, we use the Wiener filtering step (not to be confused with [1]) :

$$\boldsymbol{x}^0 = \frac{1}{\alpha}\mathcal{F}^{-1}\left\{\frac{\overline{\mathcal{F}(\boldsymbol{h})}\mathcal{F}(\boldsymbol{y})}{1/\alpha + |\mathcal{F}(\boldsymbol{h})|^2}\right\}, \tag{16}$$

where the constant factor $1/\alpha$ in the denominator represents the inverse of the signal-to-noise ratio of the blurred measurements. Note that this step can be derived as an ℓ_2 regularized solution of the deconvolution problem as well.

Hyperparameters: The parameters used in updates (10a), (10c) – ρ_1, ρ_2 are changed for each iteration and determined in one-shot by the blurring kernel $\boldsymbol{h}$ and photon level α as they control the degradation of the image. The kernel $\boldsymbol{h}$ is used as input to 4 convolutional layers, flattened to a vector of length 1024. Along the with the photon level α, the flattened vector is used as an input to a 3-layer fully connected network which output two set of vectors i.e. $\{\rho_1^1, \rho_1^2, \ldots, \rho_1^K\}$ and $\{\rho_2^1, \rho_2^2, \ldots, \rho_2^K\}$. We refer the readers to the supplementary document for further architectural details.

Note that there is no ground-truth assumed for parameters ρ_1, ρ_2 as the hyperparameter network described above is trained simultaneously as rest of the parameters of the network.

Denoiser: For the denoiser used in (13), we use the architecture provided in [46] which introduces skip connections in a U-Net architecture known as ResUNet. Like a standard U-Net, there are four downsampling operations followed by 4 upsampling operations with skip connections between the upsampling and downsampling operators. The denoiser weights are shared across the unrolling iterations instead of different set of weights for each iteration. For further details of the architecture we refer the readers to [46] or the supplementary document. Note that in our implementation of the architecture, we do not concatenate the denoiser input $\widetilde{\boldsymbol{z}}^k$ with a noise level.

IV. Experiments

A. Training

We train the network described in section III using ℓ_1-loss function. We use images from the Flickr2K [57] dataset to train

the network. The dataset contains a total of 2650 images of which we partition using a $80/20$ split for training and validation. All images are converted to gray-scale, scaled to a size of 256×256, and are blurred using motion kernels generated from [58] and Gaussian blur kernels. Due to memory limits of GPU, random patches of size 128×128 were cropped and used as inputs for the network during training.

For training, a combination of 60 motion kernels generated from [58] and 10 isotropic gaussian blur kernels with σ varying from $[0.1, 2.5]$ were used. All the kernels were pre-generated prior to training and were randomly selected during training. Entries of the blur kernel are non-negative and sum to 1. Photon Shot noise is synthetically added to the blurred image according to (2). The photon level α is uniformly sampled from the range $[1, 60]$.

The inputs to the network consist of the blurred and corrupted image $\boldsymbol{y}$, the normalized blur kernel $\boldsymbol{h}$, and the photon noise level α. The output from the network is the reconstructed image $\boldsymbol{x}^K$ where K denotes the number of iterations for which the scheme is unrolled for. We set the the number of iterations in our implementation to $K = 8$. Using the ℓ_1-loss function, we train the network with Adam optimizer [60] using a learning rate 1×10^{-4} and batch size of 5 for 100 epochs. All the parameters of the network are initialized using Xavier initialization [61] and is implemented in Pytorch 1.7.0. For training, we use an NVIDIA Titan Xp GP102 GPU and it takes approximately 20 hours for training to complete.

B. Choice of Deblurring Methods for Comparison

Before describing the results of quantitative evaluation, we briefly discuss the other deblurring approaches we compare our method with. The methods, namely **RGDN** [2], **DWDN** [1], **DPIR** [35], and **PURE-LET** [13], were chosen because they give state-of-the-art results on the deblurring problem *and* because they represent different contemporary approaches to solving the non-blind deconvolution problem.

RGDN (Recurring Gradient Descent Network) is an unrolled optimization method. More specifically, the authors take the deconvolution cost function $||\boldsymbol{y} - \boldsymbol{k} * \boldsymbol{x}||^2 + \Omega(\boldsymbol{x})$ and provide a gradient descent iterative scheme for it. The second term in the cost functions represents image prior and the corresponding gradient term $\nabla\Omega(\boldsymbol{x})$ is estimated using a convolutional neural network and the network, after being unrolled for fixed iterations, is trained end-to-end.

Deep-Weiner Deconvolution (DWDN) can be viewed as a hybrid deconvolution/denoising method. As a U-Net denoiser converts an image into a smaller feature space and then reconstructs the image using a decoder, DWDN first extracts features, performs Weiner deconvolution in that feature space, and then followed by decoding to a clean image. Through this architecture choice, they are able to perform denoising through the encoder-decoder structure but also deblur the image using Weiner deconvolution.

DPIR (Deep Plug-and-Play Image Restortation) uses a pre-trained denoiser in a half-quadratic splitting scheme and represents a state-of-the-art method which can be used for general purpose linear inverse problems like super-resolution and deblurring. Like our approach, it also boils down to a iterative series of denoising and deblurring steps.

PURE-LET (Poisson Unbiased Risk Estimate - Linear Expansion of Thresholds) proposes the solutions as a linear combination of basis function whose weights are determined by minimizing the unbiased estimate of the mean squared loss under given noise conditions. While not a deep-learning method, it performs surprisingly competitively and can incorporate both Poisson shot noise and Gaussian read noise explicitly.

Unrolled network have received a growing interest in the signal and image processing community [51]. However, the vast majority of the methods are based on the Gaussian likelihood [50], [62]. Since our problem is Poisson, comparing our method against those Gaussian-based unrolled networks is a mismatch. Replacing the Gaussian likelihood with a Poisson likelihood would resolve this issue, but doing so would require a redesign of the unrolled network which is exactly the purpose of this paper. As such, the most relevant evaluation would be a comparison between the various two-way splitting and the three-way splitting strategies which will be shown in Section IV.D. Other unrolled methods such as [47], [48] are designed for blind deconvolution. The work we consider here is non-blind deconvolution.

C. Quantitative Evaluation

The results are summarized in Fig. 6. We evaluate our method using synthetically generated noisy blurred images on 100 images from the BSDS300 dataset [63], from now on referred to as *BSD100*. We evaluate the performance on different photon levels ($\alpha = 5, 10, 20, 40$) representing various levels of degradation in terms of signal-to-noise ratio. We test the methods for different blur kernels - specifically 4 isotropic Gaussian kernels, 4 anisotropic Guassian kernels, and 4 motion kernels, as illustrated in Fig. 7. Note that the top-left kernel's width is very small - this can be viewed as an identity operator and hence equivalent to evaluating the method's performance on denoising (as opposed to deblurring).

As described in the previous subsection, we compare our method with the following deblurring methods - **RGDN**, **PURE-LET**, **DWDN**, and **DPIR**. Different features of the above-mentioned deconvolution approaches have been summarized in Table II for reader's convenience. For the sake of a fair comparison, the end-to-end trainable methods RGDN and DWDN were retrained using the same procedure as that of our method.

In addition to the BSD100 dataset, we also evaluated these methods on the blurring dataset provided in Levin et al. [59]. This dataset contains a set of 32 blurred images generated by blurring 4 different clean images by 8 different motion kernels. We synthetically corrupt the blurred images with Poisson noise at different illumination levels.

The results for these evaluations are provided in Table (III) and Fig. 6. For qualitative comparison on grayscale and colour reconstructions, one can refer to Fig. 8. On the BSDS100 dataset, our method outperforms the competing methods on

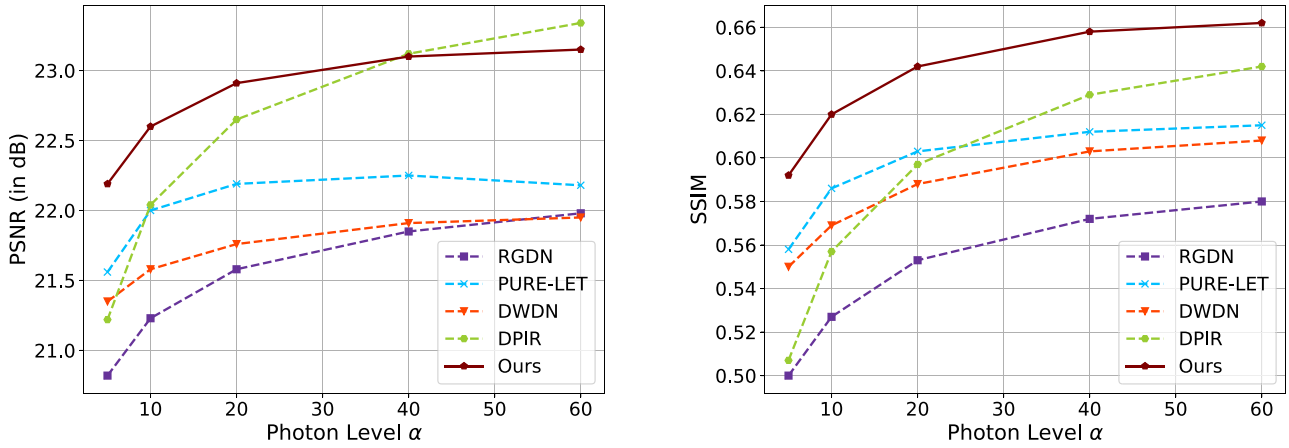


Fig. 6. Quantitative evaluation. Comparison of PSNR and SSIM of the different methods on Levin et al. dataset [59]. The dataset consists of 32 blurred images generated by blurring 4 images by 8 motion kernels and average PSNR/SSIM for all images and kernels plotted for different photon levels. The images were corrupted by Poisson noise at photon levels $\alpha = 5, 10, 20, 40$ and 60.

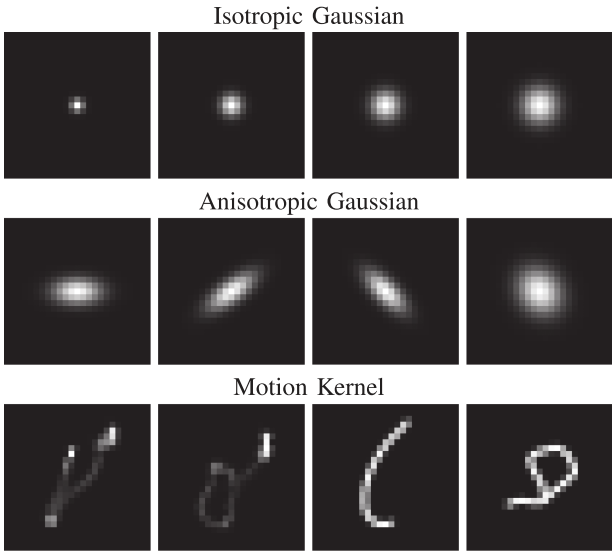


Fig. 7. Kernels used for evaluation on BSD100 dataset.

TABLE II
DIFFERENT FEATURES OF METHODS USED IN THIS PAPER FOR POISSON DEBLURRING. WE CLASSIFY THE METHODS BASED ON THREE CRITERIA - ITERATIVE/NON-ITERATIVE, END-TO-END TRAINABILITY AND WHETHER THE MODEL EXPLICITLY INCORPORATES THE FACT THAT THE IMAGES ARE CORRUPTED BY POISSON SHOT NOISE

Method	Iterative?	End-to-End Trainable?	Handles Poisson Noise?
RGDN [2]	✓	✓	×
PURE-LET [13]	×	×	✓
DWDN [1]	×	✓	×
DPIR [35]	✓	×	×
PhD-Net (Ours)	✓	✓	✓

all blurring kernels and illumination levels. For the dataset by Levin et al., we outperform the other methods except DPIR at photon level $\alpha = 40$. On both datasets, we observe that the gap between conventional deblurring and our method decreases as the illumination levels increase. This is because as the mean of a Poisson random variable starts increasing, the probability distribution function resembles that of a Gaussian. Therefore, the conventional deblurring methods which are designed for Gaussian noise show improved performance.

D. Comparison Between 2-Operator and 3-Operator Splitting

As explained in Section III-B, conventional Plug-and-Play using two-operator splitting is not suitable for algorithm unrolling. The proposed three-operator splitting enables algorithm unrolling because every iterative step is differentiable. It is this end-to-end training that allows us to a better performance. In this experiment, we perform an ablation study to quantify the performance gain through different combinations of unrolling and training.

In Fig. 9, we show the reconstruction performance of three schemes on the BSD100 dataset: (a) conventional two-operator splitting PnP using FFDNet denoiser as described in Section III-B (b) an alternate three-operator splitting formulation using FFDNet as described in Section III-C and (c) the proposed unrolled version of the scheme described in Section III-C. The results show that the two iterative schemes (a) and (b) perform similarly. However, training the proposed algorithm unrolling achieves a consistent performance gain of more than 1 dB across all photon levels.

When implementing the conventional PnP in (a), we use the approach from [18] and solve the $\boldsymbol{x}$-update (8a) using a L-BFGS solver [55]. Like the original implementation, we use a surrogate cost function to approximate the near zero entries with a quadratic approximation to avoid the singularities in the original cost function. A pretrained DnCNN [64] for noise level $\sigma = 15/255$ was used for the $\boldsymbol{z}$-update (8b). For the three-operator splitting scheme in (b), the same denoiser was used. To ensure a fair comparison, in the proposed fixed iteration unrolled network, we replace the ResUNet denoiser with a DnCNN and train it using the method described in Section IV-A. Further details about the experiment are provided in the supplementary document.

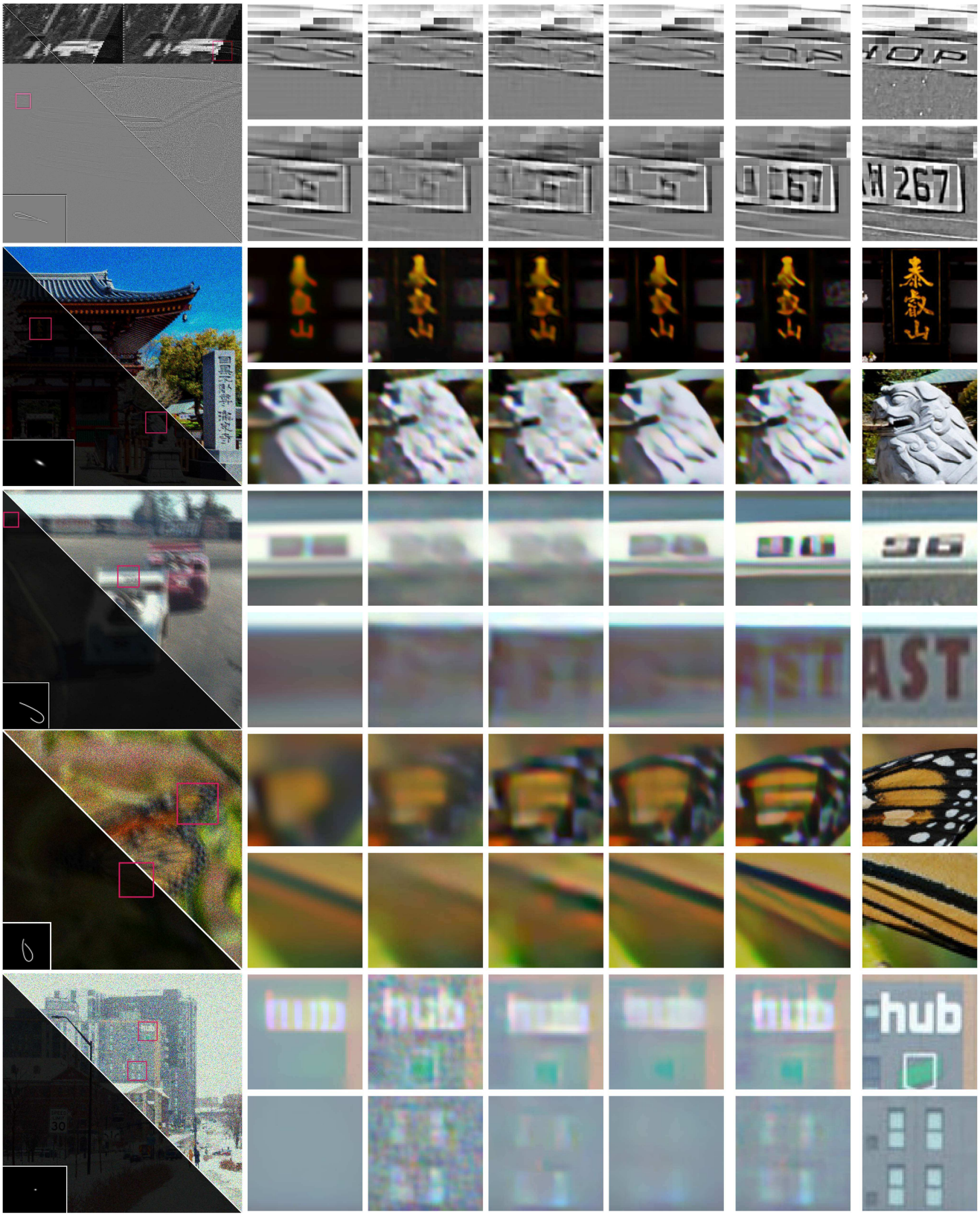


Fig. 8. Qualitative Evaluation on synthetic images. We compare the performance of the proposed method with competing methods on synthetic grayscale and color images.

TABLE III
COMPARISON OF PROPOSED METHOD WITH OTHER COMPETING APPROACHES ON BSD100 DATASET

Photon Level	Kernel		RGDN [2]	PURE-LET [13]	DWDN [1]	DPIR [35]	PhD-Net (Ours)
$\alpha = 5$	Isotropic Gaussian	PSNR (dB)	21.77	22.78	22.50	22.33	**23.46**
		SSIM	0.440	0.502	0.493	0.431	**0.531**
	Anisotropic Gaussian	PSNR (dB)	21.62	22.22	22.19	21.92	**22.70**
		SSIM	0.427	0.463	0.464	0.409	**0.491**
	Motion	PSNR (dB)	21.14	21.49	21.54	21.35	**22.12**
		SSIM	0.377	0.419	0.413	0.377	**0.433**
$\alpha = 10$	Isotropic Gaussian	PSNR (dB)	22.57	23.54	22.86	23.17	**24.24**
		SSIM	0.491	0.549	0.527	0.476	**0.576**
	Anisotropic Gaussian	PSNR (dB)	22.30	22.81	22.56	22.60	**23.28**
		SSIM	0.466	0.501	0.494	0.448	**0.525**
	Motion	PSNR (dB)	21.51	22.07	21.94	21.98	**22.80**
		SSIM	0.399	0.454	0.443	0.411	**0.475**
$\alpha = 20$	Isotropic Gaussian	PSNR (dB)	23.11	24.27	23.16	23.98	**24.96**
		SSIM	0.528	0.594	0.558	0.522	**0.621**
	Anisotropic Gaussian	PSNR (dB)	22.78	23.34	22.86	23.20	**23.83**
		SSIM	0.494	0.536	0.522	0.485	**0.557**
	Motion	PSNR (dB)	21.82	22.70	22.27	22.65	**23.47**
		SSIM	0.418	0.494	0.475	0.448	**0.515**
$\alpha = 40$	Isotropic Gaussian	PSNR (dB)	23.47	25.00	23.35	24.76	**25.68**
		SSIM	0.555	0.638	0.582	0.569	**0.663**
	Anisotropic Gaussian	PSNR (dB)	23.10	23.82	23.10	23.74	**24.36**
		SSIM	0.515	0.569	0.545	0.520	**0.589**
	Motion	PSNR (dB)	22.07	23.38	22.52	23.36	**24.20**
		SSIM	0.436	0.538	0.502	0.488	**0.564**

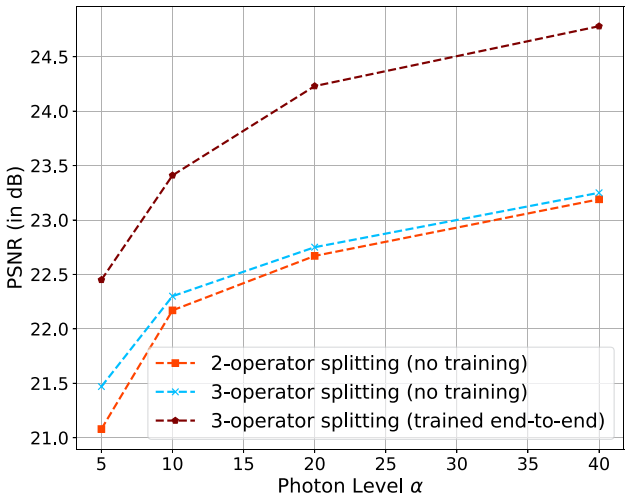


Fig. 9. Ablation study to quantify significance of algorithm unrolling. We evaluate the following three schemes on the BSD100 dataset **(a)** conventional PnP (two-operator splitting) with a DnCNN denoiser. **(b)** alternate PnP (three-operator splitting) with a DnCNN denoiser. **(c)** proposed fixed iteration unrolled network using a DnCNN denoiser. The results of this experiment show that the significant improvement is achieved due to the network unrolling.

E. Color Reconstruction

The focus of this paper is image deblurring. We acknowledge that most image sensors today acquire color images using the color filter arrays. However, adding the deblurring task with demosaicking is substantially beyond the scope of this paper. Even for demosaicking without any blur, the shot noise requires customized design, e.g., [65]. Therefore, color images shown in this paper were processed individually for each color channel and then fused using an off-the-shelf demosaicking algorithm. While this approach is sub-optimal, our real image experiments show that the performance is acceptable.

V. REAL SENSOR DATA

Unlike conventional deblurring problems where datasets are widely available, photon-limited deblurring data is not easy to collect. In this section we report our efforts in collecting a new dataset for evaluating low-light deblurring algorithms.

A. Photon-Limited Deblurring Dataset

We collect shot-noise corrupted and blurred images using a digital single lens reflex (DSLR) camera. The DLSR is hand-held to generate motion blur. A Dell 24-inch monitor, pointing towards the region of interest, was used as a programmable illumination source to control the photon level α. A light-meter is placed in the scene to measure the photon flux level.

Image Capture: We use an Canon EOS Rebel T6i camera to capture the images with exposure time of 30 ms and aperture $f/5.0$. The ISO was set to the highest possible value of 12800 to maximize the internal gain of the sensor and hence minimize the quantization effects of the analog-to-digital convertor (ADC).

(a) Experimental setup, well illuminated scene (b) Real capture

Fig. 10. Experimental setup. For evaluation of the proposed method on real images, we collect noisy and blurred images using a DSLR as shown in the setup shown above. To capture a single degraded image, we reduce the illumination to a level that shot noise becomes visible. We blur image using camera shake. For the blur kernel, each scene contains a point source and the corresponding motion kernels can be visualized in Fig. 11.

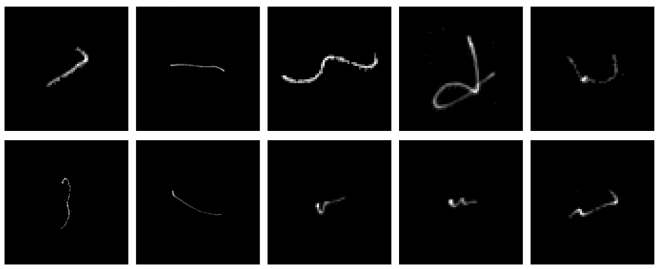


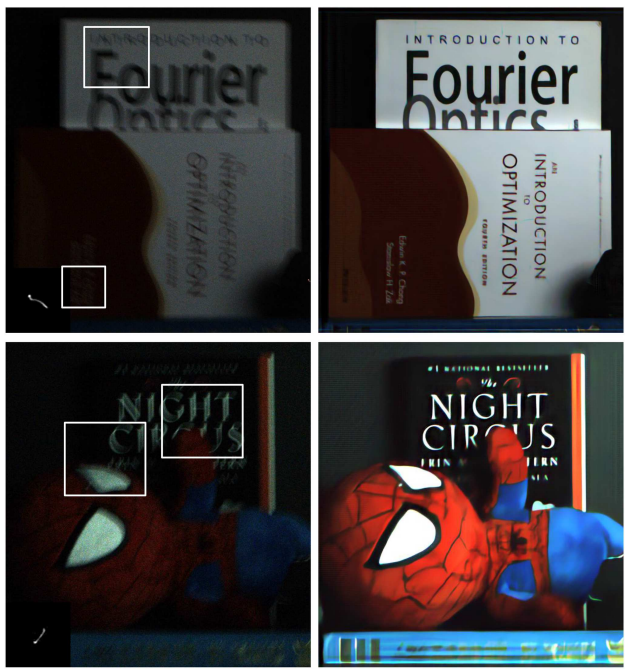
Fig. 11. Real kernels generated by our optical experiment setup.

The same scene was captured using different illumination levels and correspondingly different motion blur kernels. The raw image files were used for image processing instead of the compressed JPG files.

Generating Blur: To capture the blur kernel along with the image, we place a point source in each scene (see bottom right of middle image in Fig. 10). The point source is created by placing an LED behind a black screen with a $30\mu m$ pinhole. The strength of the point source is maximized to ensure the kernel is not corrupted with shot noise without saturation of pixel values. Some example kernels collected through this process can be visualized in Fig. 11.

Photon Level: The illumination of the scenes varies between 1-5 Lux, as measured by the light-meter shown in Fig. 10. To maximize the amount of photons captured, the aperture is kept as large as possible. However shot noise is still present due to the relatively short exposure time. The estimated average photons-per-pixel (ppp) varied from 5-60.

Generating Ground Truths: For quantitative evaluation, we also provide the ground truth for each noisy blurred image. For each noisy image corrupted by motion and noise - we place the camera on a tripod and capture 10 frames of the scene under the same illumination and camera settings. The frames, captured without any blur due to camera shake, are averaged to reduce the shot noise as much as possible. These images serve as ground truth when evaluating the performance of reconstruction methods using PSNR/SSIM.



(a) Real input (b) Processed

Fig. 12. Proposed method on real data. For a qualitative comparison of other deblurring approaches on these images, refer to Fig. 13.

B. Reconstruction From Real Data

Pre-processing: To reconstruct the images using our network, we first need to convert it into the format representing the number of photons captured from the raw sensor values. The raw digital data ($\boldsymbol{y}_{\text{raw}}$) from the. RAW file is presented using a 14-bit value.

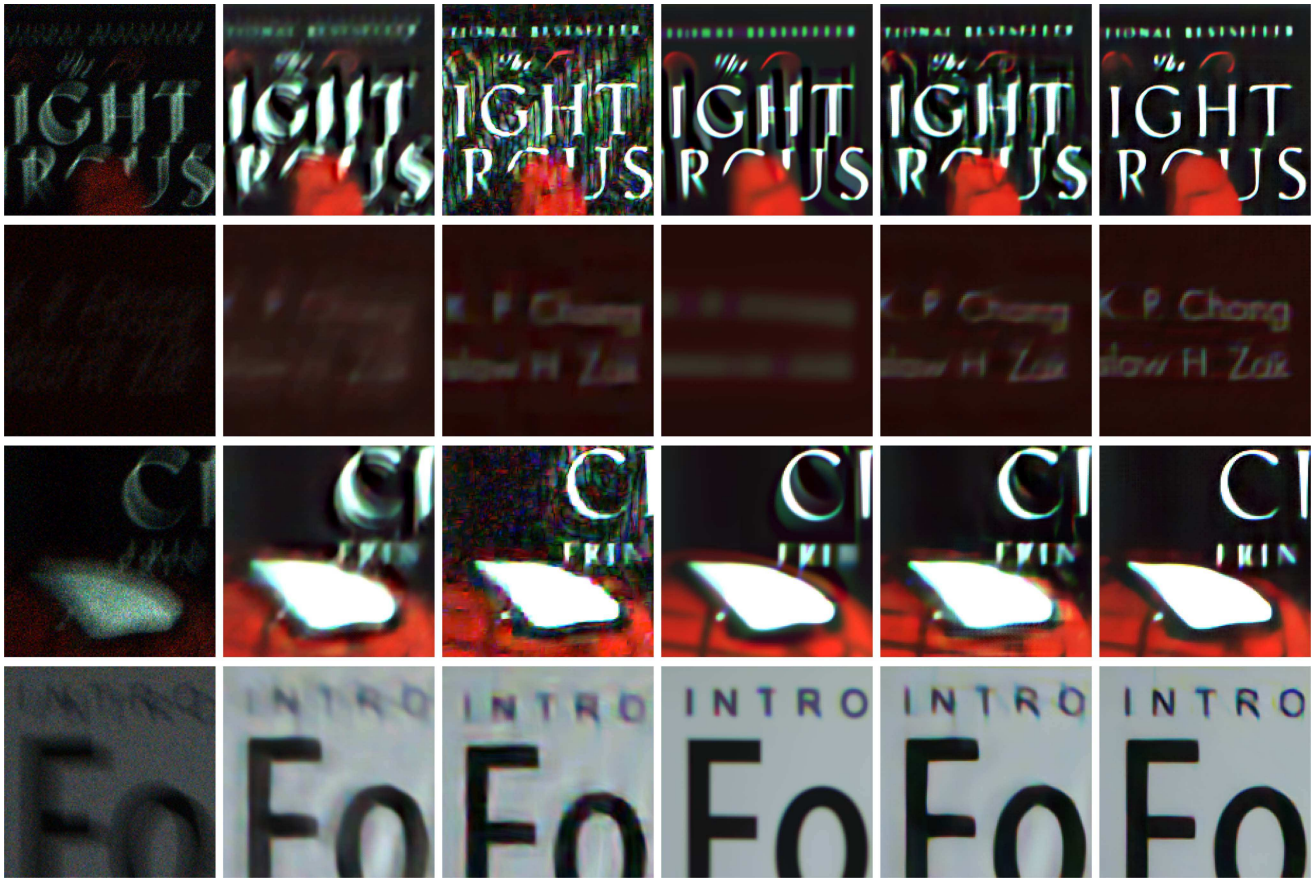


Fig. 13. Qualitative Comparison We look at zoomed in regions of the reconstructed images from Fig. 12 using competing methods. The average PSNR and SSIM evaluated on the given patches is provided at the bottom. From visual inspection one can see that our method is able to recover finer details compared to other methods. Note that in the first row, the DPIR output may look qualitatively similar to our result. This is because the former often tends to blur out images for a "cleaner" looking image as observed in the second row of zoomed in reconstructions.

To convert the 14-bit format to the number of photons, we use the following linear transform

$$\boldsymbol{y}_i = \frac{\boldsymbol{y}_{i\,\mathrm{raw}} - b}{G}, \tag{17}$$

where b represents the zero-level offset of the camera which can be obtained from the metadata of the image. RAW file and is set equal to $b = 2047$. G represents the gain factor between the digital output of the sensor and the actual electrons collected by the sensor. This gain is calculated from the camera data available at [66]. Specifically, we look at the read noise of the camera in terms of digital numbers and electrons. The ratio of these two data will give the gain G. For Canon EOS Rebel T6i, at ISO 12800, the gain is estimated to be $G \approx 71$.

Our reconstruction results are shown in Fig. 12. We also compare reconstructions using proposed method with other contemporary deblurring methods (RGDN, PURE-LET, DPIR and DWDN) in Fig. 13. Through a visual inspection, one can conclude that our method is able to reconstruct finer details from the noisy and blurred image while leaving behind fewer artifacts.

Quantitative Evaluation: For evaluation of metrics such as PSNR and SSIM, we register the ground truth to the reconstruction using homography transformation to account for the differences in camera positions. The average PSNR and SSIM on the real datset for the proposed and competing methods are reported in Table IV. We outperform the second-best competing methods, i.e. [1], by 0.6 dB in terms of PSNR and by 0.005 in terms of SSIM. As shown in Fig. 13, when evaluating SSIM on a few patches containing text, the gap between our method and [1] becomes much wider.

TABLE IV
PSNR (IN DB) AND SSIM EVALUATED ON REAL DATASET OF 30 IMAGES

Method	RGDN [2]	PURE-LET [13]	DPIR [35]	DWDN [1]	**PhD-Net (Ours)**
PSNR	19.80	20.88	22.09	22.85	**23.48**
SSIM	0.476	0.501	0.548	0.561	**0.566**

VI. CONCLUSION AND FUTURE WORK

In this paper, we formulated the photon-limited deblurring problem as a Poisson inverse problem. We presented an end-to-end trainable solution using a algorithm unrolling technique. We performed extensive numerical experiments to compare

our approach with other existing state-of-the-art non-blind deblurring approaches and demonstrated how our method can be applied to real sensor data. Even though the present solution is focused on image deblurring, it can be easily extended to other photon-limited inverse problems such as compressive sensing, lensless imaging, and super-resolution.

The algorithm presented in this paper can be used to reconstruct a single clean image from multiple blurred images. This would allow us to take advantage of the temporal redundancy which would be necessary to obtain a meaningful clean signal in much challenging scenarios (e.g. photon level $\alpha \leq 5$). Another interesting but challenging problem which can be attempted using the framework is *low-light blind deconvolution* i.e. recovering the clean image and blur kernel simultaneously from blurred images corrupted with photon shot noise.

REFERENCES

[1] J. Dong, S. Roth, and B. Schiele, "Deep wiener deconvolution: Wiener meets deep learning for image deblurring," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1048–1059, 2020.

[2] D. Gong, Z. Zhang, Q. Shi, A. van den Hengel, C. Shen, and Y. Zhang, "Learning deep gradient descent optimization for image deconvolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5468–5482, Dec. 2020.

[3] J. Kruse, C. Rother, and U. Schmidt, "Learning to push the limits of efficient FFT-based image deconvolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4586–4594.

[4] T. Eboli, J. Sun, and J. Ponce, "End-to-end interpretable learning of non-blind image deblurring," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 314–331.

[5] Y. Nan, Y. Quan, and H. Ji, "Variational-EM-based deep learning for noise-blind image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3626–3635.

[6] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.

[7] H. Wang and P. C. Miller, "Scaled heavy-ball acceleration of the richardson-lucy algorithm for 3-D microscopy image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 848–854, Feb. 2014.

[8] J.-L. Starck and F. Murtagh, *Astronomical Image and Data Analysis*. Berlin, Germany: Springer Science & Business Media, 2007.

[9] J. W. Goodman, *Statistical Optics*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015.

[10] Y. Chi, A. Gnanasambandam, V. Koltun, and S. H. Chan, "Dynamic low-light imaging with quanta image sensors," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 122–138.

[11] A. Gnanasambandam, O. Elgendy, J. Ma, and S. H. Chan, "Megapixel photon-counting color imaging using quanta image sensor," *Opt. Exp.*, vol. 27, no. 12, pp. 17298–17310, 2019.

[12] C. Li, X. Qu, A. Gnanasambandam, O. A. Elgendy, J. Ma, and S. H. Chan, "Photon-limited object detection using non-local feature matching and knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3976–3987.

[13] J. Li, F. Luisier, and T. Blu, "PURE-LET image deconvolution," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 92–105, Jan. 2018.

[14] W. H. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Amer.*, vol. 62, no. 1, pp. 55–59, 1972.

[15] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astronomical J.*, vol. 79, 1974, Art. no. 745.

[16] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 945–948.

[17] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.

[18] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *J. Vis. Commun. Image Representation*, vol. 41, pp. 96–108, 2016.

[19] M. Bertero, P. Boccacci, G. Desidera, and G. Vicidomini, "Image deblurring with poisson data: From cells to galaxies," *Inverse Problems*, vol. 25, no. 12, 2009, Art. no. 123006.

[20] L. A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Imag.*, vol. 1, no. 2, pp. 113–122, Oct. 1982.

[21] N. Dey et al., "Richardson–lucy algorithm with total variation regularization for 3d confocal microscope deconvolution," *Microsc. Res. Techn.*, vol. 69, no. 4, pp. 260–266, 2006.

[22] M. Laasmaa, M. Vendelin, and P. Peterson, "Application of regularized richardson–lucy algorithm for deconvolution of confocal microscopy images," *J. Microsc.*, vol. 243, no. 2, pp. 124–140, 2011.

[23] M. A. Figueiredo and J. M. Bioucas-Dias, "Restoration of poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.

[24] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "This is SPIRAL-TAP: Sparse Poisson intensity reconstruction algorithms–theory and practice," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1084–1096, Mar. 2011.

[25] R. D. Nowak and E. D. Kolaczyk, "A statistical multiscale framework for poisson inverse problems," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1811–1825, May 2000.

[26] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.

[27] F. Xue, F. Luisier, and T. Blu, "Multi-wiener SURE-LET deconvolution," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1954–1968, May 2013.

[28] F. J. Anscombe, "The transformation of poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[29] M. Makitalo and A. Foi, "Optimal inversion of the anscombe transformation in low-count poisson image denoising," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 99–109, Jan. 2010.

[30] F. Luisier, C. Vonesch, T. Blu, and M. Unser, "Fast interscale wavelet denoising of poisson-corrupted images," *Signal Process.*, vol. 90, no. 2, pp. 415–427, 2010.

[31] B. Zhang, J. M. Fadili, and J.-L. Starck, "Wavelets, ridgelets, and curvelets for poisson noise removal," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1093–1108, Jul. 2008.

[32] L. Azzari and A. Foi, "Variance stabilization for noisy estimate combination in iterative poisson denoising," *IEEE signal Process. Lett.*, vol. 23, no. 8, pp. 1086–1090, Aug. 2016.

[33] S. Sreehari et al., "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, Apr. 2016.

[34] R. Ahmad et al., "Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery," *IEEE signal Process. Mag.*, vol. 37, no. 1, pp. 105–116, Jan. 2020.

[35] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3929–3938.

[36] T. He, Y. Sun, B. Chen, J. Qi, W. Liu, and J. Hu, "Plug-and-play inertial forward–backward algorithm for Poisson image deconvolution," *J. Electron. Imag.*, vol. 28, no. 4, 2019, Art. no. 0 43020.

[37] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, Dec. 2017.

[38] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 395–408, Mar. 2019.

[39] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM J. Imag. Sci.*, vol. 11, no. 3, pp. 2001–2020, Mar. 2018.

[40] S. H. Chan, "Performance analysis of plug-and-play ADMM: A graph signal processing perspective," *IEEE Trans. Comput. Imag.*, vol. 5, no. 2, pp. 274–286, Feb. 2019.

[41] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5546–5557.

[42] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[43] R. Cohen, M. Elad, and P. Milanfar, "Regularization by denoising via fixed-point projection (RED-PRO)," *SIAM J. Imag. Sci.*, vol. 14, no. 3, pp. 1374–1406, 2021.

[44] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comput. Imag.*, vol. 5, no. 1, pp. 52–67, Mar. 2019.

[45] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
[46] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3217–3226.
[47] Y. Li, M. Tofighi, V. Monga, and Y. C. Eldar, "An algorithm unrolling approach to deep image deblurring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7675–7679.
[48] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 666–681, 2020.
[49] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 10–18.
[50] R. Liu, S. Cheng, L. Ma, X. Fan, and Z. Luo, "Deep proximal unrolling: Algorithmic framework, convergence analysis and applications," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5013–5026, Oct. 2019.
[51] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Feb. 2021.
[52] D. Gilton, G. Ongie, and R. Willett, "Deep equilibrium architectures for inverse problems in imaging," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 1123–1133, 2021.
[53] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.
[54] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Boston, MA, USA: Now Publishers Inc, 2011.
[55] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, 1995.
[56] M. A. Figueiredo and J. M. Bioucas-Dias, "Deconvolution of poissonian images using variable splitting and augmented lagrangian optimization," in *Proc. IEEE/SP 15th Workshop Stat. Signal Process.*, 2009, pp. 733–736.
[57] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 136–144.
[58] G. Boracchi and A. Foi, "Modeling the performance of image restoration from motion blur," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3502–3517, Aug. 2012.
[59] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1964–1971.
[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations (Poster)*, 2022. Accessed: Sep. 20, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
[61] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[62] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, "Unrolled optimization with deep priors," 2017, *arXiv:1705.08041*.
[63] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE 8th Int. Conf. Comput. Vis.* 2001, pp. 416–423, vol. 2, 2001.
[64] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[65] O. A. Elgendy, A. Gnanasambandam, S. H. Chan, and J. Ma, "Low-light demosaicking and denoising for small pixels using learned frequency selection," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 137–150, 2021.
[66] "Photons to photos," Accessed: Sep. 09, 2021. [Online]. Available: https://www.photonstophotos.net/

Yash Sanghvi (Student Member, IEEE) received the B.Tech. and M.Tech degree in electrical engineering from the Indian Institute of Technology, Bombay, Mumbai, India, in 2018. He is currently working toward the Ph.D. degree with the Purdue University, West Lafayette, IN, USA. He joined Purdue University in 2019, where he is a Graduate Research Assistant of electrical and computer engineeringwith the Intelligent Imaging Lab. From 2018 to 2019, he was working on the inverse scattering problem as a Project Scientist with the Indian Institute of Technology Madras, Chennai, India. His research interests include inverse problems, computational imaging, and deep learning.

Abhiram Gnanasambandam (Member, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Madras, Chennai, India, in 2017, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2022. His Ph.D. dissertation dealt with computational imaging, image processing, deep learning, and computer vision. He is currently a Senior Engineer with Samsung Research America, Mountain View, CA, USA.

Stanley H. Chan (Senior Member, IEEE) received the B.Eng. degree (with first-class Hons.) in electrical engineering from the University of Hong Kong, Hong Kong, in 2007, and the M.A. degree in mathematics and the Ph.D. degree in electrical engineering from the University of California at San Diego, San Diego, CA, USA, in 2009 and 2011, respectively. From 2012 to 2014, he was a Postdoctoral Research Fellow with Harvard University, Cambridge, MA, USA. He joined Purdue University, West Lafayette, IN, USA, in 2014, where he is currently an Elmore Associate Professor of electrical and computer engineering, with a joint appointment with the Department of Statistics. Dr. Chan is the author of a popular undergraduate textbook *Introduction to Probability for Data Science*, Michigan Publishing 2021. The book is part of the University of Michigan's free textbook initiative to disseminate high-quality educational materials to students and families around the world. Since the launch of the book in Fall 2021, the book has saved more than $3 million USD for families globally. His research interests include single-photon imaging, imaging through atmospheric turbulence, and computational photography. Dr. Chan was the recipient of the Best Paper Award of IEEE International Conference on Image Processing 2016, IEEE Signal Processing Cup 2016 Second Prize, Purdue College of Engineering Exceptional Early Career Teaching Award 2019, Purdue College of Engineering Outstanding Graduate Mentor Award 2016, and Eta Kappa Nu (Beta Chapter) Outstanding Teaching Award 2015. Since 2018, he has been an Associate Editor for the IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING, where he is recognized as an outstanding Editorial Board Member in 2021. He was also an Associate Editor for the OSA *Optics Express* in 2016-2018, and an Elected Member of the IEEE Signal Processing Society Technical Committee in Computational Imaging in 2015-2020.