Deep Plug-and-play Nighttime Non-blind Deblurring with Saturated Pixel Handling Schemes
Supplemental Material

Hung-Yu Shu Yi-Hsien Lin Yi-Chang Lu
National Taiwan University
{r09943021, d06943006, yiclu}@ntu.edu.tw

1. Overview

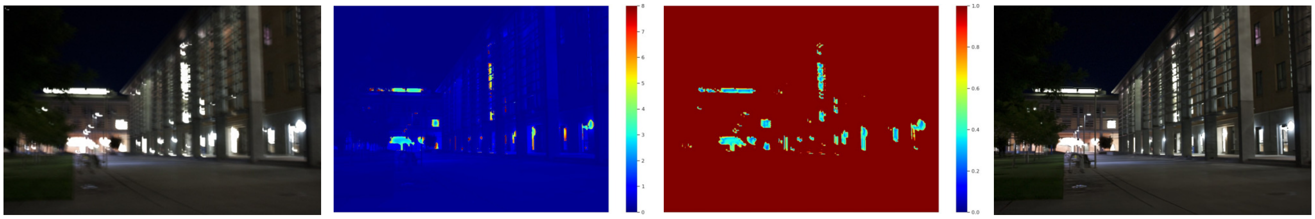
The supplementary material is organized as follows. First, the mask visualization is shown in Section. 2. The Derivation of the proximal operators is presented in Section. 3, followed by the Derivation of the nighttime non-blind deblur algorithm in Section. 4. The implementation details of the unified algorithm are elaborated in Section. 5, and more analyses are presented in Section. 6, 7, 8, 9 and 10. Finally, more experimental results on the benchmark datasets and real-world blurry images are given in Section. 11.

2. Mask Visualization

This part presents the visualizations of the pixel stretching mask and the image segment mask, followed by the analysis of the cross-channel operation.

2.1. Pixel Stretching Mask

The pixel stretching mask M serves as a clipping function to make the value of the blurred image within the sensor range, and an example of the visualization is illustrated in Fig. 8. Fig. 8 (a) is the blurred image. Here, we visualize the deblurred result and the pixel stretching mask of the G channel in Fig. 8 (b) and (c), where the heat maps are used to include the pixel values out of the sensor range. Similar results can be obtained in R and B channels as well. Fig. 8 (b) shows that the pixel values of saturated pixels can be out of the sensor range $[0, 1]$ during deconvolution. This kind of outlier often causes severe artifacts, so the pixel stretching mask M (consisting of M_G, M_R, and M_B) is used to adjust the pixel value of blurred images. Fig. 8 (d) is the deblurred result after applying the pixel stretching mask M.

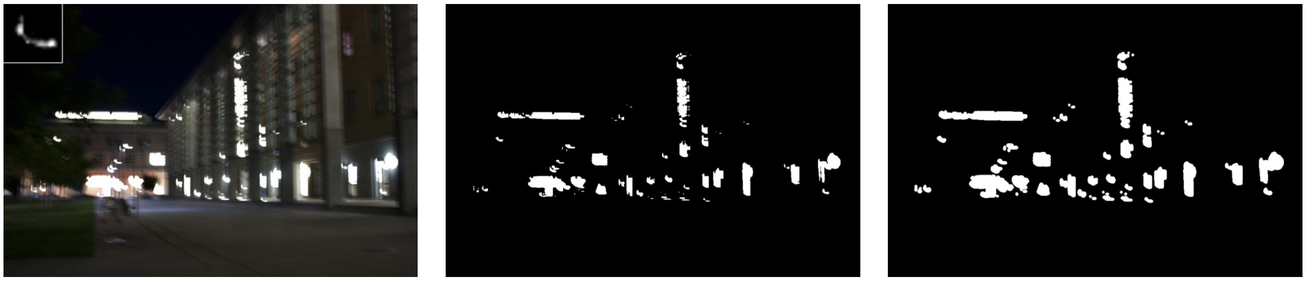


(a) B (b) I_G (c) M_G (d) I

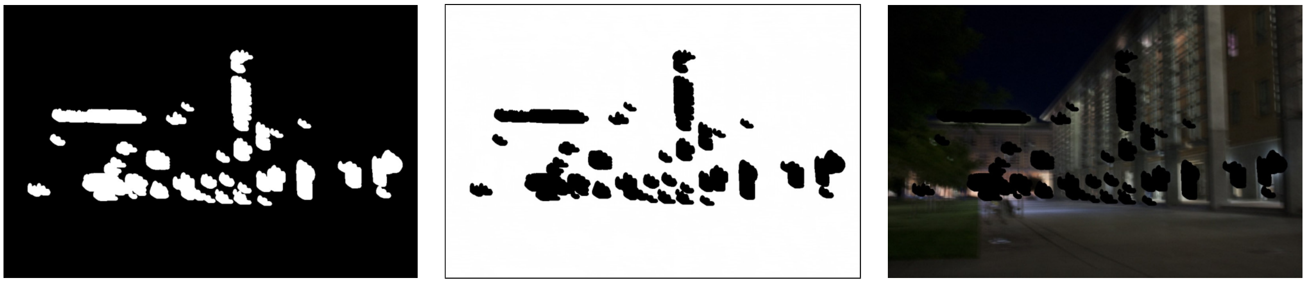
Fig. 8. Visualization of the deblurred result and the pixel stretching mask. (a) Blurred image B. (b) The deblurred result of the G channel I_G. (c) The pixel stretching mask M_G corresponding to (b). (d) The deblurred result I.

2.2. Image Segment Mask

As suggested by [15], we can decouple saturated pixels from the blurred image through a few operations. For instance, Fig. 9 (a) is the blurry image B with saturated pixels as well as a provided kernel K, and Fig. 9 (b) is the corresponding hard threshold mask of Fig. 9 (a). Empirically, we set the hard threshold value as 0.9 as suggested in [15] and the parameter is not sensitive to the exact value. During deconvolution, non-saturated pixels near the saturation region can already be affected by the saturated pixels, so image dilation is adopted for the saturation region $\mathbb{S}$ to reduce impacts on boundaries. Fig. 9 (c) is the corresponding dilated image of Fig. 9 (b) and Fig. 9 (d) is the corresponding dilated mask $M_{\mathbb{S}}$ of (c), where the binary structuring element is in the shape of the blur kernel. We can effectively segment saturated pixels from the non-saturation region through the above operations. The corresponding image segment mask $M_{\mathbb{U}}$ as well as the non-saturation region $M_{\mathbb{U}} \circ B$ extracted with $M_{\mathbb{U}}$ are demonstrated in Fig. 9 (e) and Fig. 9 (f).



(a) Blurry input (b) Hard threshold mask for B (c) Dilated image



(d) $M_{\mathbb{S}}$ (e) $M_{\mathbb{U}}$ (f) $M_{\mathbb{U}} \circ B$

Fig. 9. Illustration of segment masks. (a) Blurred image B with saturated pixels and a provided kernel. (b) The corresponding hard threshold mask of (a). (c) The corresponding dilated image of (b). (d) The corresponding dilated mask $M_{\mathbb{S}}$ of (c) with the structuring element in the shape of blur kernel. (e) The image segment mask $M_{\mathbb{U}}$. (f) Non-saturation regions $M_{\mathbb{U}} \circ B$ extracted with the mask $M_{\mathbb{U}}$.

2.3. Analysis of the Cross-Channel Operation

The proposed segmentation process considers cross-channel consistency to refine the mask. Fig. 10 (a) is the mask $M_{\mathbb{U}}$ without considering cross-channel consistency, and Fig. 10 (c) is the deblurred result corresponding to Fig. 10 (a). Take the color of Fig. 10 (a) as an example: the white pixels in the mask $M_{\mathbb{U}}$ indicate that all RGB channels are not saturated, and the black pixels in the mask represent that all channels are saturated. By the same token, the cyan pixels in the mask mean that both B and G channels are not saturated. From Fig. 10 (c), this color mismatch will generate some color artifacts during deconvolution. Hence, we incorporate cross-channel consistency to refine the mask, and the refined mask and the corresponding deblurred result is visualized in Fig. 10 (b) and Fig. 10 (d).

3. Derivation of the Proximal Operators

In this part, we go through the derivation of the proximal operators to the Poisson data prior and the indicator function. Refer to [14] for more details.

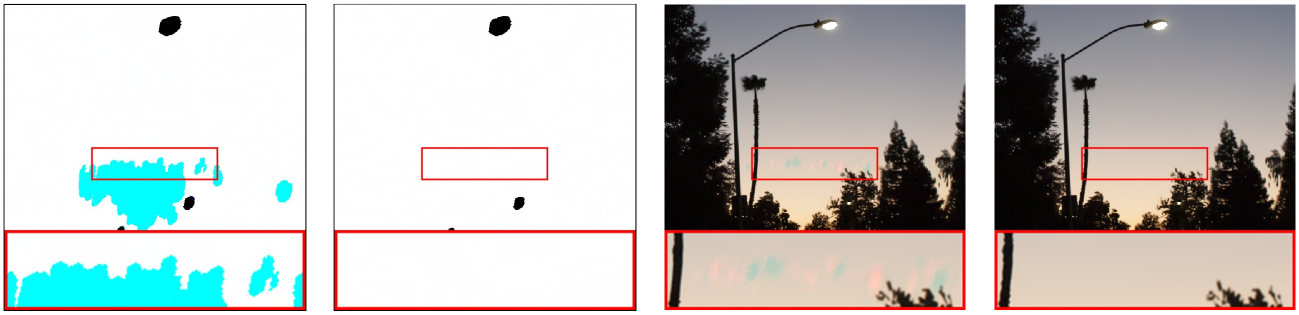

(a) $\boldsymbol{M}_{\mathbb{U}}$ w/o cross-channel (b) $\boldsymbol{M}_{\mathbb{U}}$ w/ cross-channel (c) Deblurred result w/ (a) (d) Deblurred result w/ (b)

Fig. 10. The cross-channel consistency removes color artifacts from outliers of the image segment mask.

3.1. Proximal operator for the Poisson data prior

We can define the proximal operator for the maximum likelihood estimator of the Poisson distribution as follows:

$$\begin{aligned}\boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{V},\boldsymbol{B}) &= \arg\min_{\boldsymbol{Z}} \mathcal{P}(\boldsymbol{Z},\boldsymbol{B}) + \frac{\rho}{2}||\boldsymbol{Z}-\boldsymbol{V}||_2^2 \\ &= \arg\min_{\boldsymbol{Z}} \sum_i Z_i - B_i\log(Z_i) + \log(B_i!) + \frac{\rho}{2}||\boldsymbol{Z}-\boldsymbol{V}||_2^2 \\ &= \arg\min_{\boldsymbol{Z}} \sum_i Z_i - B_i\log(Z_i) + \frac{\rho}{2}||\boldsymbol{Z}-\boldsymbol{V}||_2^2 \\ &= \arg\min_{\boldsymbol{Z}} \mathbf{1}^T\boldsymbol{Z} - \boldsymbol{B}^T log(\boldsymbol{Z}) + \frac{\rho}{2}||\boldsymbol{Z}-\boldsymbol{V}||_2^2.\end{aligned} \tag{19}$$

To solve the proximal operator, we can equate the first-order derivative of the Eq. (19) to zero as follows:

$$\mathbf{1} - \boldsymbol{diag}(\boldsymbol{Z}^{-1})\boldsymbol{B} + \rho(\boldsymbol{Z}-\boldsymbol{V}) = 0. \tag{20}$$

When looking at the Eq. (20) with respect to an individual element Z_i in $\boldsymbol{Z}$, we have:

$$1 - \frac{B_i}{Z_i} + \rho\left(Z_i - V_i\right) = 0. \tag{21}$$

The solution to Eq. (21) is the root of a quadratic equation so that we can define the proximal operator as:

$$\boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{V},\boldsymbol{B}) = -\left(\frac{1-\rho\boldsymbol{V}}{2\rho}\right) \pm \sqrt{\left(\frac{1-\rho\boldsymbol{V}}{2\rho}\right)^2 + \frac{\boldsymbol{B}}{\rho}}. \tag{22}$$

Due to the camera sensor range of image pixel values, we are only interested in the positive one. Thus Eq. (22) can be reduced to:

$$\boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{V},\boldsymbol{B}) = -\left(\frac{1-\rho\boldsymbol{V}}{2\rho}\right) + \sqrt{\left(\frac{1-\rho\boldsymbol{V}}{2\rho}\right)^2 + \frac{\boldsymbol{B}}{\rho}}, \tag{23}$$

which is a pixel-wise operator without any iterations, and ρ is a parameter for the ADMM algorithm.

3.2. Proximal operator for indicator function

We can define the proximal operator of the non-negative constraint as:

$$\begin{aligned}\boldsymbol{Prox}_{\mathbb{I}_{\mathbb{R}_+}}(\boldsymbol{V}) &= \arg\min_{\boldsymbol{Z}} \mathbb{I}_{\mathbb{R}_+}(\boldsymbol{Z}) + \frac{\rho}{2}||\boldsymbol{Z}-\boldsymbol{V}||_2^2 \\ &= \arg\min_{\boldsymbol{Z}\in\mathbb{R}_+} \frac{\rho}{2}||\boldsymbol{Z}-\boldsymbol{V}||_2^2 \\ &= \begin{cases} V_i, & V_i > 0 \\ 0, & V_i \le 0 \end{cases} \\ &= max(V_i, 0),\end{aligned} \tag{24}$$

where $\boldsymbol{Prox}_{\mathbb{I}_{\mathbb{R}_+}}$ is a projection operator onto the convex set $\mathbb{R}_+$ and V_i denotes individual element in $\boldsymbol{V}$.

4. Derivation of Nighttime Non-blind Deblur Algorithm

As shown in Fig. 3 in our manuscript, we can suppress ringing artifacts by separating saturated pixels from the non-saturation region with the image segment mask $M_{\mathbb{U}}$. Hence, we formulate two optimization problems: one for both saturation and non-saturation regions (i.e., Eq. (7) in our manuscript) and the other for refining the non-saturation regions suffering severe ringing (i.e., Eq. (8) in our manuscript). Next, we will go through the derivation of the two optimization problems.

4.1. Derivation of an optimization problem with both saturated pixels and non-saturated pixels

Based on Eq. (9) in our manuscript, we reformulate Eq. (7) in our manuscript as follows:

$$\arg\min_{\boldsymbol{I}} \mathbf{1}^T(\boldsymbol{KI}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{KI}) + \log((\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})!) + \lambda R(\boldsymbol{I}), \tag{25}$$

where $\boldsymbol{I}$, $\boldsymbol{B}$, $\boldsymbol{M}$ are the vectorized form of I, B, M and $\boldsymbol{K}$ is the Toeplitz matrix of K. Because $\log((\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})!)$ is a constant in the minimization problem with respect to $\boldsymbol{I}$, we drop it and re-write Eq. (25) as follows:

$$\arg\min_{\boldsymbol{I}} \mathbf{1}^T(\boldsymbol{KI}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{KI}) + \lambda R(\boldsymbol{I}). \tag{26}$$

By introducing the auxiliary variables $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$, we formulate Eq. (26) as the standard form of ADMM:

$$\begin{aligned} &\arg\min_{\boldsymbol{I}} \mathbf{1}^T(\boldsymbol{Z_1}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_1}) + \lambda R(\boldsymbol{Z_2}), \\ &\qquad\qquad \text{s.t.} \underbrace{\begin{bmatrix} \boldsymbol{K} \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{I} - \underbrace{\begin{bmatrix} \boldsymbol{Z_1} \\ \boldsymbol{Z_2} \end{bmatrix}}_{\boldsymbol{Z}} = 0, \end{aligned} \tag{27}$$

where $\boldsymbol{A}$ denotes the combined matrix by stacking the component matrices and $\boldsymbol{Z}$ denotes the auxiliary variable.

Then, we formulate the augmented Lagrangian of Eq. (27) as follows:

$$\begin{aligned} L_\rho(\boldsymbol{I}, \boldsymbol{Z}, \boldsymbol{Y}) = \mathbf{1}^T(\boldsymbol{Z_1}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_1}) + \lambda R(\boldsymbol{Z_2}) \\ + \boldsymbol{Y}^T(\boldsymbol{AI} - \boldsymbol{Z}) \\ + \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{Z}||_2^2, \end{aligned} \tag{28}$$

where $\boldsymbol{Y}$ denotes the Lagrangian multiplier and ρ denotes the penalty parameter.

To combine the linear and quadratic terms in Eq. (28), $\boldsymbol{U} = \frac{\boldsymbol{Y}}{\rho}$ is defined as the scaled Lagrangian multiplier and we have:

$$\begin{aligned} \boldsymbol{Y}^T(\boldsymbol{AI} - \boldsymbol{Z}) + \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{Z}||_2^2 &= \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{Z}||_2^2 + \boldsymbol{Y}^T(\boldsymbol{AI} - \boldsymbol{Z}) + \frac{\rho}{2}||\frac{\boldsymbol{Y}}{\rho}||_2^2 - \frac{\rho}{2}||\frac{\boldsymbol{Y}}{\rho}||_2^2 \\ &= \frac{\rho}{2}(||\boldsymbol{AI} - \boldsymbol{Z}||_2^2 + 2(\frac{\boldsymbol{Y}}{\rho})^T(\boldsymbol{AI} - \boldsymbol{Z}) + ||\frac{\boldsymbol{Y}}{\rho}||_2^2) - \frac{\rho}{2}||\frac{\boldsymbol{Y}}{\rho}||_2^2 \\ &= \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{Z} + \frac{\boldsymbol{Y}}{\rho}||_2^2 - \frac{\rho}{2}||\frac{\boldsymbol{Y}}{\rho}||_2^2 \\ &= \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{Z} + \boldsymbol{U}||_2^2 - \frac{\rho}{2}||\boldsymbol{U}||_2^2. \end{aligned} \tag{29}$$

Based on Eq. (29), we reformulate Eq. (28) as the scaled augmented Lagrangian:

$$\begin{aligned} L_\rho(\boldsymbol{I}, \boldsymbol{Z}, \boldsymbol{U}) = \mathbf{1}^T(\boldsymbol{Z_1}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_1}) + \lambda R(\boldsymbol{Z_2}) \\ + \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{Z} + \boldsymbol{U}||_2^2 \\ - \frac{\rho}{2}||\boldsymbol{U}||_2^2, \end{aligned} \tag{30}$$

where $\boldsymbol{U}$ denotes the scaled Lagrangian multiplier, and there are three subproblems, $\boldsymbol{Z_1}$, $\boldsymbol{Z_2}$ and $\boldsymbol{I}$. Now, we separately solve the three subproblems, $\boldsymbol{Z_1}$, $\boldsymbol{Z_2}$ and $\boldsymbol{I}$ as follows:

$\boldsymbol{Z_1}$-subproblem

$$\arg\min_{\boldsymbol{Z_1}} \mathbf{1}^T(\boldsymbol{Z_1}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_1}) + \frac{\rho}{2}||\boldsymbol{Z_1} - (\boldsymbol{KI} + \boldsymbol{U_1})||_2^2 \tag{31}$$

Replace $\boldsymbol{KI} + \boldsymbol{U_1}$ with $\boldsymbol{V_1}$ and formulate Eq. (31) as the proximal operator of the Poisson data term in Eq. (23):

$$\begin{aligned} &\arg\min_{\boldsymbol{Z_1}} \mathbf{1}^T(\boldsymbol{Z_1}) - (\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_1}) + \frac{\rho}{2}||\boldsymbol{Z_1} - \boldsymbol{V_1}||_2^2 \\ &= \boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{V_1}, \frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B}) \\ &= -\left(\frac{1 - \rho\boldsymbol{V_1}}{2\rho}\right) + \sqrt{\left(\frac{1 - \rho\boldsymbol{V_1}}{2\rho}\right)^2 + \frac{\frac{\mathbf{1}}{\boldsymbol{M}} \circ \boldsymbol{B}}{\rho}}, \end{aligned} \tag{32}$$

where $\boldsymbol{V_1} = \boldsymbol{KI} + \boldsymbol{U_1}$.

$\boldsymbol{Z_2}$-subproblem

$$\arg\min_{\boldsymbol{Z_2}} \lambda R(\boldsymbol{Z_2}) + \frac{\rho}{2}||\boldsymbol{Z_2} - (\boldsymbol{I} + \boldsymbol{U_2})||_2^2 \tag{33}$$

Replace $\boldsymbol{I} + \boldsymbol{U_2}$ with $\boldsymbol{V_2}$, and we can formulate the solution to $\boldsymbol{Z_2}$-subproblem as a CNN denoiser. Here, we plug and play the pre-trained DRUNet [17] into our algorithm, and Eq. (33) can be written as:

$$\begin{aligned} &\arg\min_{\boldsymbol{Z_2}} \lambda R(\boldsymbol{Z_2}) + \frac{\rho}{2}||\boldsymbol{Z_2} - \boldsymbol{V_2}||_2^2 \\ &= \boldsymbol{Prox}_{\lambda R,\rho}(\boldsymbol{V_2}) \\ &= \boldsymbol{Denoiser}(\boldsymbol{V_2}, \sqrt{\frac{\lambda}{\rho}}), \end{aligned} \tag{34}$$

where $\boldsymbol{V_2} = \boldsymbol{I} + \boldsymbol{U_2}$.

$\boldsymbol{I}$-subproblem

$$\begin{aligned} &\arg\min_{\boldsymbol{I}} \frac{\rho}{2}||\boldsymbol{AI} - (\boldsymbol{Z} - \boldsymbol{U})||_2^2 \\ &= \arg\min_{\boldsymbol{I}} || \underbrace{\begin{bmatrix} \boldsymbol{K} \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{I} - (\boldsymbol{Z} - \boldsymbol{U})||_2^2 \end{aligned} \tag{35}$$

Replace $\boldsymbol{Z} - \boldsymbol{U}$ with $\boldsymbol{V}$, and we can get the closed-form solution for the quadratic subproblem as:

$$\begin{aligned} &\arg\min_{\boldsymbol{I}} \frac{\rho}{2}||\boldsymbol{AI} - \boldsymbol{V}||_2^2 \\ &= \arg\min_{\boldsymbol{I}} || \underbrace{\begin{bmatrix} \boldsymbol{K} \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{I} - \underbrace{\begin{bmatrix} \boldsymbol{V_1} \\ \boldsymbol{V_2} \end{bmatrix}}_{\boldsymbol{V}})||_2^2 \\ &= (\boldsymbol{K}^T\boldsymbol{K} + \mathbf{1})^{-1}(\boldsymbol{K}^T\boldsymbol{V_1} + \boldsymbol{V_2}). \end{aligned} \tag{36}$$

Because $\boldsymbol{K}$ is a Toeplitz matrix of the blur kernel, the closed-form solution of Eq. (37) can be derived using Fourier transforms as well as element-wise multiplications and divisions with the circular boundary conditions [16].

$$\begin{aligned} &\boldsymbol{Quad}_{FFT}(\boldsymbol{V}, \boldsymbol{K}) \\ = \quad &\mathcal{F}^{-1}(\frac{\mathcal{F}(K)^*\mathcal{F}(V_1) + \mathcal{F}(V_2)}{(|\mathcal{F}(K)|^2 + 1)} \end{aligned} \tag{37}$$

Following the padding paradigm of [11], we smoothly expand the blurred image to a large tile to satisfy the circular boundary conditions to reduce boundary artifacts in deblurred results. After solving three subproblems, we can incorporate the proposed pixel stretching mask M into the ADMM flow as shown in Algorithm 2.

Algorithm 2 ADMM algorithm for an optimization problem deblurring both saturated and non-saturated pixels

```
Input: $\boldsymbol{B}, \boldsymbol{K}, \lambda, \rho, T$.
Initialization: $\boldsymbol{I}^{(0)} = \boldsymbol{Z_1}^{(0)} = \boldsymbol{Z_2}^{(0)} = \boldsymbol{U_1}^{(0)} = \boldsymbol{U_2}^{(0)} = \boldsymbol{B}$
  for $t \leftarrow 0$ to $T-1$ do
    $\boldsymbol{I}^{(t+1)} = \boldsymbol{Quad}_{FFT}(\boldsymbol{Z}^{(t)} - \boldsymbol{U}^{(t)}, \boldsymbol{K})$
    $\boldsymbol{M}^{(t+1)} = \boldsymbol{\Psi}_M(\boldsymbol{I}^{(t+1)}, \boldsymbol{K})$
    $\boldsymbol{Z_1}^{(t+1)} = \boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{K}\boldsymbol{I}^{(t+1)} + \boldsymbol{U_1}^{(t)}, \frac{\boldsymbol{1}}{\boldsymbol{M}^{(t+1)}} \circ \boldsymbol{B})$
    $\boldsymbol{Z_2}^{(t+1)} = \boldsymbol{Prox}_{\lambda R,\rho}(\boldsymbol{I}^{(t+1)} + \boldsymbol{U_2}^{(t)}, \sqrt{\lambda/\rho})$
    $\boldsymbol{U_1}^{(t+1)} = \boldsymbol{U_1}^{(t)} + \boldsymbol{K}\boldsymbol{I}^{(t+1)} - \boldsymbol{Z_1}^{(t+1)}$
    $\boldsymbol{U_2}^{(t+1)} = \boldsymbol{U_2}^{(t)} + \boldsymbol{I}^{(t+1)} - \boldsymbol{Z_2}^{(t+1)}$
    $\boldsymbol{Z}^{(t+1)} = \left[\boldsymbol{Z_1}^{(t+1)}; \boldsymbol{Z_2}^{(t+1)}\right]$
    $\boldsymbol{U}^{(t+1)} = \left[\boldsymbol{U_1}^{(t+1)}; \boldsymbol{U_2}^{(t+1)}\right]$
  end for
Output: $\boldsymbol{I}^{(T)}$
```

4.2. Derivation of an optimization problem addressing the refinement of non-saturated pixels

Based on Eq. (9) in our manuscript, we reformulate Eq. (8) in our manuscript as follows:

$$\arg\min_{\boldsymbol{I}_{\mathbb{U}}} \boldsymbol{1}^T(\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{K}\boldsymbol{I}_{\mathbb{U}}) - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B})^T log\,(\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{K}\boldsymbol{I}_{\mathbb{U}}) + \lambda R(\boldsymbol{I}_{\mathbb{U}}) + I_{\mathbb{R}_+}(\boldsymbol{I}_{\mathbb{U}}), \tag{38}$$

where $\boldsymbol{I}_{\mathbb{U}}$, $\boldsymbol{B}$, $\boldsymbol{M}_{\mathbb{U}}$ are the vectorized form of $I_{\mathbb{U}}, B, M_{\mathbb{U}}$ and $\boldsymbol{K}$ is Toeplitz matrix of K. Different from Eq. (26), we add a non-negative constraint because $M_{\mathbb{U}}$ is a binary mask. That is to say, as $M_{\mathbb{U}}$ goes 0, the solution to $I_{\mathbb{U}}$ should stay positive. Therefore, we add a hard constraint to ensure the pixel value in $\mathbb{R}_+$.

By introducing the auxiliary variables $\boldsymbol{Z_2}$, $\boldsymbol{Z_3}$ and $\boldsymbol{Z_4}$, we formulate Eq. (38) as the standard form of ADMM:

$$\begin{aligned} &\arg\min_{\boldsymbol{I}_{\mathbb{U}}} \boldsymbol{1}^T(\boldsymbol{Z_3}) - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_3}) + \lambda R(\boldsymbol{Z_2}) + I_{\mathbb{R}_+}(\boldsymbol{Z_4}), \\ &\text{s.t.} \underbrace{\begin{bmatrix} \mathbb{I} \\ diag(\boldsymbol{M}_{\mathbb{U}})\boldsymbol{K} \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{A}_{\mathbb{U}}} \boldsymbol{I}_{\mathbb{U}} - \underbrace{\begin{bmatrix} \boldsymbol{Z_2} \\ \boldsymbol{Z_3} \\ \boldsymbol{Z_4} \end{bmatrix}}_{\boldsymbol{Z}_{\mathbb{U}}} = 0, \end{aligned} \tag{39}$$

where $diag$ denotes the diagonal operator, $\boldsymbol{A}_{\mathbb{U}}$ denotes the combined matrix by stacking the component matrices and $\boldsymbol{Z}_{\mathbb{U}}$ denotes the auxiliary variable. Then, we formulate the augmented Lagrangian of Eq. (39) as follows:

$$\begin{aligned} L_\rho(\boldsymbol{I}_{\mathbb{U}}, \boldsymbol{Z}_{\mathbb{U}}, \boldsymbol{Y}_{\mathbb{U}}) = \boldsymbol{1}^T(\boldsymbol{Z_3}) - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_3}) + \lambda R(\boldsymbol{Z_2}) + I_{\mathbb{R}_+}(\boldsymbol{Z_4}) \\ + \boldsymbol{Y}_{\mathbb{U}}^T(\boldsymbol{A}_{\mathbb{U}}\boldsymbol{I}_{\mathbb{U}} - \boldsymbol{Z}_{\mathbb{U}}) \\ + \frac{\rho}{2}||\boldsymbol{A}_{\mathbb{U}}\boldsymbol{I}_{\mathbb{U}} - \boldsymbol{Z}_{\mathbb{U}}||_2^2, \end{aligned} \tag{40}$$

where $\boldsymbol{Y}_{\mathbb{U}}$ denotes the Lagrangian multiplier. Based on Eq. (29), we reformulate Eq. (40) as the scaled augmented Lagrangian:

$$\begin{aligned} L_\rho(\boldsymbol{I}_{\mathbb{U}}, \boldsymbol{Z}_{\mathbb{U}}, \boldsymbol{U}_{\mathbb{U}}) = \boldsymbol{1}^T(\boldsymbol{Z_3}) - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_3}) + \lambda R(\boldsymbol{Z_2}) + I_{\mathbb{R}_+}(\boldsymbol{Z_4}) \\ + \frac{\rho}{2}||\boldsymbol{A}_{\mathbb{U}}\boldsymbol{I}_{\mathbb{U}} - \boldsymbol{Z}_{\mathbb{U}} + \boldsymbol{U}_{\mathbb{U}}||_2^2 \\ - \frac{\rho}{2}||\boldsymbol{U}_{\mathbb{U}}||_2^2, \end{aligned} \tag{41}$$

where $\boldsymbol{U}_{\mathbb{U}}$ denotes the scaled Lagrangian multiplier. Now, we separately solve the four subproblems, $\boldsymbol{Z_2}$, $\boldsymbol{Z_3}$, $\boldsymbol{Z_4}$ and $\boldsymbol{I}_{\mathbb{U}}$ as follows:

$\boldsymbol{Z_2}$-subproblem

$$\arg\min_{\boldsymbol{Z_2}} \lambda R(\boldsymbol{Z_2}) + \frac{\rho}{2}||\boldsymbol{Z_2} - (\boldsymbol{I}_{\mathbb{U}} + \boldsymbol{U_2})||_2^2 \tag{42}$$

Similar to the derivation of Eq. (33), we formulate the solution to $\boldsymbol{Z_2}$-subproblem as a plug-and-play CNN denoiser. Here, we plug and play the pre-trained DRUNet [17] into our algorithm and Eq. (42) can be written as:

$$\begin{aligned}
&\arg\min_{\boldsymbol{Z_2}} \lambda R(\boldsymbol{Z_2}) + \frac{\rho}{2}||\boldsymbol{Z_2} - \boldsymbol{V_2}||_2^2 \\
&= \boldsymbol{Prox}_{\lambda R,\rho}(\boldsymbol{V_2}) \\
&= \boldsymbol{Denoiser}(\boldsymbol{V_2}, \sqrt{\frac{\lambda}{\rho}}),
\end{aligned} \tag{43}$$

where $\boldsymbol{V_2} = \boldsymbol{I}_{\mathbb{U}} + \boldsymbol{U_2}$.

$\boldsymbol{Z_3}$-subproblem

$$\arg\min_{\boldsymbol{Z_3}} \boldsymbol{1}^T(\boldsymbol{Z_3}) - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_3}) + \frac{\rho}{2}||\boldsymbol{Z_3} - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{K}\boldsymbol{I}_{\mathbb{U}} + \boldsymbol{U_3})||_2^2 \tag{44}$$

Similar to the derivation of Eq. (31), we formulate Eq. (44) as a proximal operator of the Poisson data prior in Eq. (23) as:

$$\begin{aligned}
&\arg\min_{\boldsymbol{Z_3}} \boldsymbol{1}^T(\boldsymbol{Z_3}) - (\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B})^T log\,(\boldsymbol{Z_3}) + \frac{\rho}{2}||\boldsymbol{Z_3} - \boldsymbol{V_3}||_2^2 \\
&= \boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{V_3}, \boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B}) \\
&= -\left(\frac{1-\rho\boldsymbol{V_3}}{2\rho}\right) + \sqrt{\left(\frac{1-\rho\boldsymbol{V_3}}{2\rho}\right)^2 + \frac{\boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{B}}{\rho}},
\end{aligned} \tag{45}$$

where $\boldsymbol{V_3} = \boldsymbol{M}_{\mathbb{U}} \circ \boldsymbol{K}\boldsymbol{I}_{\mathbb{U}} + \boldsymbol{U_3}$.

$\boldsymbol{Z_4}$-subproblem

$$\arg\min_{\boldsymbol{Z_4}} \mathbb{I}_{\mathbb{R}_+}(\boldsymbol{Z_4}) + \frac{\rho}{2}||\boldsymbol{Z_4} - (\boldsymbol{I}_{\mathbb{U}} + \boldsymbol{U_4})||_2^2 \tag{46}$$

We can formulate the solution to $\boldsymbol{Z_4}$-subproblem as a point wise projection operator in Eq. (24):

$$\begin{aligned}
&\arg\min_{\boldsymbol{Z_4}} \mathbb{I}_{\mathbb{R}_+}(\boldsymbol{Z_4}) + \frac{\rho}{2}||\boldsymbol{Z_4} - \boldsymbol{V_4}||_2^2 \\
&= \boldsymbol{Prox}_{\mathbb{I}_{\mathbb{R}_+}}(\boldsymbol{V_4}) \\
&= max(V_{4i}, 0),
\end{aligned} \tag{47}$$

where $\boldsymbol{V_4} = \boldsymbol{I}_{\mathbb{U}} + \boldsymbol{U_4}$ and V_{4i} denotes individual element in $\boldsymbol{V_4}$.

$\boldsymbol{I}_{\mathbb{U}}$-subproblem

$$\begin{aligned}
&\arg\min_{\boldsymbol{I}_{\mathbb{U}}} \frac{\rho}{2}||\boldsymbol{A}_{\mathbb{U}}\boldsymbol{I}_{\mathbb{U}} - (\boldsymbol{Z}_{\mathbb{U}} - \boldsymbol{U}_{\mathbb{U}})||_2^2 \\
&= \arg\min_{\boldsymbol{I}_{\mathbb{U}}} || \underbrace{\begin{bmatrix} \mathbb{I} \\ diag(\boldsymbol{M}_{\mathbb{U}})\boldsymbol{K} \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{A}_{\mathbb{U}}} I_{\mathbb{U}} - (\boldsymbol{Z}_{\mathbb{U}} - \boldsymbol{U}_{\mathbb{U}})||_2^2
\end{aligned} \tag{48}$$

With $\boldsymbol{V}_{\mathbb{U}} = \boldsymbol{Z}_{\mathbb{U}} - \boldsymbol{U}_{\mathbb{U}}$, we can get the closed-form solution for the quadratic subproblem:

$$\begin{aligned} & \arg\min_{\boldsymbol{I}_{\mathbb{U}}} \frac{\rho}{2} ||\boldsymbol{A}_{\mathbb{U}} \boldsymbol{I}_{\mathbb{U}} - \boldsymbol{V}_{\mathbb{U}}||_2^2 \\ = \; & \arg\min_{\boldsymbol{I}_{\mathbb{U}}} || \underbrace{\begin{bmatrix} \mathbb{I} \\ diag(\boldsymbol{M}_{\mathbb{U}})\boldsymbol{K} \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{A}_{\mathbb{U}}} I_{\mathbb{U}} - \underbrace{\begin{bmatrix} \boldsymbol{V_2} \\ \boldsymbol{V_3} \\ \boldsymbol{V_4} \end{bmatrix}}_{\boldsymbol{V}_{\mathbb{U}}})||_2^2 \\ = \; & (\boldsymbol{1} + \boldsymbol{K}^T diag(\boldsymbol{M}_{\mathbb{U}})^T diag(\boldsymbol{M}_{\mathbb{U}})\boldsymbol{K} + \boldsymbol{1})^{-1}(\boldsymbol{V_2} + \boldsymbol{K}^T diag(\boldsymbol{M}_{\mathbb{U}})^T \boldsymbol{V_3} + \boldsymbol{V_4}). \end{aligned} \tag{49}$$

Although K is a Toeplitz matrix of the blur kernel, it is coupled with a Hadamard product with $M_{\mathbb{U}}$. Hence, we use the conjugate gradient method [4] instead of the closed-form method in section 4.1. We define the solution of the conjugate gradient procedure as follows:

$$\begin{aligned} & \boldsymbol{Quad}_{CG}(\boldsymbol{M}_{\mathbb{U}}, \boldsymbol{V}, \boldsymbol{K}) \\ = \; & (\boldsymbol{1} + \boldsymbol{K}^T diag(\boldsymbol{M}_{\mathbb{U}})^T diag(\boldsymbol{M}_{\mathbb{U}})\boldsymbol{K} + \boldsymbol{1})^{-1}(\boldsymbol{V_2} + \boldsymbol{K}^T diag(\boldsymbol{M}_{\mathbb{U}})^T \boldsymbol{V_3} + \boldsymbol{V_4}). \end{aligned} \tag{50}$$

After solving the four subproblems, we can summarize these solutions with the image segment mask in Algorithm 3.

Algorithm 3 ADMM algorithm for an optimization problem addressing the refinement of non-saturated pixels

```
Input: $\boldsymbol{B}, \boldsymbol{K}, \lambda, \rho, T.$
Initialization: $\boldsymbol{I}_{\mathbb{U}}^{(0)} = \boldsymbol{Z}_k{}^{(0)} = \boldsymbol{U}_k{}^{(0)} = \boldsymbol{B}\ (k = 2, 3, 4)$
Mask Initialization: $\boldsymbol{M}_{\mathbb{U}}{}^{(0)} = \boldsymbol{\Psi}_{M_{\mathbb{U}}}(\boldsymbol{B}, \boldsymbol{K})$
  for $t \leftarrow 0$ to $T - 1$ do
    $\boldsymbol{I}_{\mathbb{U}}^{(t+1)} = \boldsymbol{Quad}_{CG}(\boldsymbol{M}_{\mathbb{U}}{}^{(t)}, \boldsymbol{Z}_{\mathbb{U}}^{(t)} - \boldsymbol{U}_{\mathbb{U}}^{(t)}, \boldsymbol{K})$
    $\boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} = \boldsymbol{\Psi}_{M_{\mathbb{U}}}(\boldsymbol{I}_{\mathbb{U}}^{(t+1)}, \boldsymbol{K})$
    $\boldsymbol{Z_2}^{(t+1)} = \boldsymbol{Prox}_{\lambda R,\rho}(\boldsymbol{I}_{\mathbb{U}}^{(t)} + \boldsymbol{U_2}^{(t)})$
    $\boldsymbol{Z_3}^{(t+1)} = \boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} \circ \boldsymbol{K}\boldsymbol{I}_{\mathbb{U}}{}^{(t+1)} + \boldsymbol{U_3}^{(t)}, \boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} \circ \boldsymbol{B})$
    $\boldsymbol{Z_4}^{(t+1)} = \boldsymbol{Prox}_{\mathbb{I}_{\mathbb{R}_+}}(\boldsymbol{I}_{\mathbb{U}}^{(t)} + \boldsymbol{U_4}^{(t)})$
    $\boldsymbol{U_2}^{(t+1)} = \boldsymbol{U_2}^{(t)} + \boldsymbol{I}_{\mathbb{U}}^{(t+1)} - \boldsymbol{Z_2}^{(t+1)}$
    $\boldsymbol{U_3}^{(t+1)} = \boldsymbol{U_3}^{(t)} + \boldsymbol{M}_{\mathbb{U}}^{(t+1)} \circ \boldsymbol{K}\boldsymbol{I}_{\mathbb{U}}^{(t)} - \boldsymbol{Z_3}^{(t+1)}$
    $\boldsymbol{U_4}^{(t+1)} = \boldsymbol{U_4}^{(t)} + \boldsymbol{I}_{\mathbb{U}}^{(t+1)} - \boldsymbol{Z_4}^{(t+1)}$
    $\boldsymbol{Z}_{\mathbb{U}}^{(t+1)} = \left[\boldsymbol{Z_2}^{(t+1)}; \boldsymbol{Z_3}^{(t+1)}; \boldsymbol{Z_4}^{(t+1)}\right]$
    $\boldsymbol{U}_{\mathbb{U}}^{(t+1)} = \left[\boldsymbol{U_2}^{(t+1)}; \boldsymbol{U_3}^{(t+1)}; \boldsymbol{U_4}^{(t+1)}\right]$
  end for
Output: $\boldsymbol{I}_{\mathbb{U}}^{(T)}$
```

5. Implementation Details of the Unified Algorithm

In this part, we go through the implementation details of the unified algorithm, Algorithm 1 in our manuscript. After we separately solve the two optimization problems, we can unify Algorithm 2 and 3 as well as the saturation awareness mechanism (SAM) into an integrated framework as Algorithm 4. The proposed SAM decides whether to enable the *highly-exposed* mode based on two criteria: whether the maximum pixel value is larger than the hard threshold β and whether the time step t is in the first half of the whole optimization process, and the condition can be formulated as:

$$max(I^{(t)}) \geq \beta \quad and \quad t \leq \frac{T}{2}, \tag{51}$$

where β denotes the hard threshold, T denotes the total number of iterations, and t denotes the time step of iterations.

Algorithm 4 Overall non-blind deblurring algorithm

```
Input: $\boldsymbol{B}, \boldsymbol{K}, \lambda^{(0)}, \rho, T, \alpha, \beta, \gamma$.
Initialization: $\tilde{S}^{(0)} = 1, \boldsymbol{I}^{(0)} = \boldsymbol{Z}^{(0)} = \boldsymbol{U}^{(0)} = \boldsymbol{B}$
  for $t \leftarrow 0$ to $T-1$ do
    $\boldsymbol{I}^{(t+1)} = \boldsymbol{Quad}_{FFT}(\boldsymbol{Z}^{(t)} - \boldsymbol{U}^{(t)}, \boldsymbol{K})$
    if highly-exposed then
      $\boldsymbol{I}_{\mathbb{U}}^{(t+1)} = \boldsymbol{Quad}_{CG}(\boldsymbol{Z}_{\mathbb{U}}^{(t)} - \boldsymbol{U}_{\mathbb{U}}^{(t)}, \boldsymbol{M}_{\mathbb{U}}{}^{(t)}, \boldsymbol{K})$
      $\boldsymbol{I}^{(t+1)} = \boldsymbol{Blend}(\boldsymbol{I}^{(t+1)}, \boldsymbol{I}_{\mathbb{U}}^{(t+1)}, \boldsymbol{M}_{\mathbb{S}}{}^{(t)})$
      $\boldsymbol{Z}_{\boldsymbol{3}}^{(t+1)} = \boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} \circ \boldsymbol{K}\boldsymbol{I}^{(t+1)} + \boldsymbol{U}_{\boldsymbol{3}}^{(t)}, \boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} \circ \boldsymbol{B})$
      $\boldsymbol{Z}_{\boldsymbol{4}}^{(t+1)} = \boldsymbol{Prox}_{\mathbb{I}_{\mathbb{R}_+}}(\boldsymbol{I}^{(t+1)} + \boldsymbol{U}_{\boldsymbol{4}}^{(t)})$
      $\boldsymbol{U}_{\boldsymbol{3}}^{(t+1)} = \boldsymbol{U}_{\boldsymbol{3}}^{(t)} + \boldsymbol{M}_{\mathbb{U}}^{(t+1)} \circ \boldsymbol{K}\boldsymbol{I}^{(t+1)} - \boldsymbol{Z}_{\boldsymbol{3}}^{(t+1)}$
      $\boldsymbol{U}_{\boldsymbol{4}}^{(t+1)} = \boldsymbol{U}_{\boldsymbol{4}}^{(t)} + \boldsymbol{I}^{(t+1)} - \boldsymbol{Z}_{\boldsymbol{4}}^{(t+1)}$
      $\boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} = \boldsymbol{\Psi}_{M_{\mathbb{U}}}(\boldsymbol{I}^{(t+1)}, \boldsymbol{K})$
      $\boldsymbol{M}_{\mathbb{S}}{}^{(t+1)} = \boldsymbol{\Psi}_{M_{\mathbb{S}}}(\boldsymbol{I}^{(t+1)}, \boldsymbol{K})$
    end if
    $\boldsymbol{M}^{(t+1)} = \boldsymbol{\Psi}_{\boldsymbol{M}}(\boldsymbol{I}^{(t+1)}, \boldsymbol{K})$
    $\boldsymbol{Z}_{\boldsymbol{1}}^{(t+1)} = \boldsymbol{Prox}_{\mathcal{P},\rho}(\boldsymbol{K}\boldsymbol{I}^{(t+1)} + \boldsymbol{U}_{\boldsymbol{1}}^{(t)}, \frac{\boldsymbol{1}}{\boldsymbol{M}^{(t+1)}} \circ \boldsymbol{B})$
    $\boldsymbol{Z}_{\boldsymbol{2}}^{(t+1)} = \boldsymbol{Prox}_{\lambda^{(t)}R,\rho}(\boldsymbol{I}^{(t+1)} + \boldsymbol{U}_{\boldsymbol{2}}^{(t)})$
    $\boldsymbol{U}_{\boldsymbol{1}}^{(t+1)} = \boldsymbol{U}_{\boldsymbol{1}}^{(t)} + \boldsymbol{K}\boldsymbol{I}^{(t+1)} - \boldsymbol{Z}_{\boldsymbol{1}}^{(t+1)}$
    $\boldsymbol{U}_{\boldsymbol{2}}^{(t+1)} = \boldsymbol{U}_{\boldsymbol{2}}^{(t)} + \boldsymbol{I}^{(\boldsymbol{t+1})} - \boldsymbol{Z}_{\boldsymbol{2}}^{(t+1)}$
    $\boldsymbol{Z}^{(t+1)} = \left[\boldsymbol{Z}_{\boldsymbol{1}}^{(t+1)}; \boldsymbol{Z}_{\boldsymbol{2}}^{(t+1)}\right]$
    $\boldsymbol{U}^{(t+1)} = \left[\boldsymbol{U}_{\boldsymbol{1}}^{(t+1)}; \boldsymbol{U}_{\boldsymbol{2}}^{(t+1)}\right]$
    if highly-exposed then
      $\boldsymbol{Z}_{\mathbb{U}}^{(t+1)} = \left[\boldsymbol{Z}_{\boldsymbol{2}}^{(t+1)}; \boldsymbol{Z}_{\boldsymbol{3}}^{(t+1)}; \boldsymbol{Z}_{\boldsymbol{4}}^{(t+1)}\right]$
      $\boldsymbol{U}_{\mathbb{U}}^{(t+1)} = \left[\boldsymbol{U}_{\boldsymbol{2}}^{(t+1)}; \boldsymbol{U}_{\boldsymbol{3}}^{(t+1)}; \boldsymbol{U}_{\boldsymbol{4}}^{(t+1)}\right]$
    end if
    if $||\Delta \boldsymbol{I}^{(t+1)}||_2 > ||\Delta \boldsymbol{I}^{(t)}||_2$ and $||\Delta \boldsymbol{Z}^{(t+1)}||_2 > ||\Delta \boldsymbol{Z}^{(t)}||_2$ and $||\Delta \boldsymbol{U}^{(t+1)}||_2 > ||\Delta \boldsymbol{U}^{(t)}||_2$ and
    $\frac{1}{\sqrt{n}}||\Delta \boldsymbol{I}^{(t+1)}||_2 + \frac{1}{\sqrt{2n}}||\Delta \boldsymbol{U}^{(t+1)}||_2 + \frac{1}{\sqrt{2n}}||\Delta \boldsymbol{Z}^{(t+1)}||_2 < \gamma / \tilde{S}^{(t)}$ then
      $\boldsymbol{Early\ Stop}$
    end if
    if $max(\boldsymbol{I}^{(t+1)}) \geq \beta$ and $t \leq \frac{T}{2}$ then
      $\boldsymbol{Enable}$ highly-exposed $\boldsymbol{mode}$
      $\boldsymbol{I}^{(t+1)} = \boldsymbol{I}_{\mathbb{U}}^{(t+1)} = \boldsymbol{Z}^{(t+1)} = \boldsymbol{U}^{(t+1)} = \boldsymbol{0}$
      $\boldsymbol{Z}_{\mathbb{U}}^{(t+1)} = \boldsymbol{U}_{\mathbb{U}}^{(t+1)} = \boldsymbol{0}$
      $\boldsymbol{M}_{\mathbb{U}}{}^{(t+1)} = \boldsymbol{\Psi}_{\boldsymbol{M}_{\mathbb{U}}}(\boldsymbol{B}, \boldsymbol{K})$
      $\boldsymbol{M}_{\mathbb{S}}{}^{(t+1)} = \boldsymbol{\Psi}_{\boldsymbol{M}_{\mathbb{S}}}(\boldsymbol{B}, \boldsymbol{K})$
    end if
    $\tilde{s}^{(t+1)} = max(\boldsymbol{I}^{(t+1)})$
    $\tilde{S}^{(t+1)} = min(2^{\tilde{s}^{(t+1)}}, \alpha)$
    $\lambda^{(t+1)} = \lambda^{(0)} \cdot \tilde{S}^{(t+1)}$
  end for
Output: $\boldsymbol{I}^{(T)}$
```

Suppose the *highly-exposed* mode is enabled. In that case, we re-initialize variables of subproblems to 0 for better performance, so the time step condition ensures sufficient iterations and avoids re-initialization in the final optimization stage. Besides, the $\boldsymbol{Z_2}$-subproblem is the same in both Algorithm 2 and 3, so we share the auxiliary variable $\boldsymbol{Z_2}$ in Algorithm 4.

After enabling the *highly-exposed* mode, we blend $\boldsymbol{I}$ and $\boldsymbol{I}_{\mathbb{U}}$ with alpha blending as proposed in [15]. Then, we use the Gaussian kernel $\boldsymbol{K_G}$ to blur the $\boldsymbol{M}_{\mathbb{S}}$ to avoid apparent boundaries between the two latent sharp images $\boldsymbol{I}$ and $\boldsymbol{I}_{\mathbb{U}}$, and the notation of alpha-blending can be written as:

$$\boldsymbol{Blend}(\boldsymbol{I}, \boldsymbol{I}_{\mathbb{U}}, \boldsymbol{M}_{\mathbb{S}}) = (\boldsymbol{M}_{\mathbb{S}} \otimes \boldsymbol{K_G}) \circ \boldsymbol{I} + (\boldsymbol{1} - \boldsymbol{M}_{\mathbb{S}} \otimes \boldsymbol{K_G}) \circ \boldsymbol{I}_{\mathbb{U}}. \tag{52}$$

Besides, since the image segment mask $M_{\mathbb{S}}$ of the observed blurred image B encloses the largest possible saturation region, we also use it as the boundary mask $M_{\mathbb{S}}^{(0)} = \boldsymbol{\Psi}_{M_{\mathbb{S}}}(B, K)$ to reject sudden changes of pixel values in the non-saturation region. Then, we can decompose I into:

$$\begin{aligned} I_{M_{\mathbb{S}}}^{(t)} &= M_{\mathbb{S}}^{(0)} \circ M_{\mathbb{S}}^{(t)} \circ I^{(t)}, \\ I_{M_{\mathbb{U}}}^{(t)} &= I^{(t)} \setminus I_{M_{\mathbb{S}}}^{(t)} = M_{\mathbb{U}}^{(t)} \circ I^{(t)}, \end{aligned} \tag{53}$$

where $M_{\mathbb{S}}^{(0)}$ denotes the binary guided mask from the observed blurred image corresponding to $\mathbb{S}^{(0)}$.

As shown in Fig. 3 of our manuscript, severe artifacts come with saturated pixels with high-intensity values. Better visual results can be obtained by choosing the larger strength for the denoiser. Hence, we adaptively adjust the regularization weight based on the maximum pixel value $\tilde{s}^{(t+1)}$ in Eq. (5) of our manuscript as follows:

$$\begin{aligned} \lambda^{(t+1)} &= \lambda^{(0)} \cdot \tilde{S}^{(t+1)} \\ \text{where} \quad \tilde{S}^{(t+1)} &= \mathit{min}(2^{\tilde{s}^{(t+1)}}, \alpha). \end{aligned} \tag{54}$$

Eq. (54) is a combination of the exponential and minimum functions where the adaptive weight $\tilde{S}^{(t+1)}$ controls the relation between the regularization weight as well as the maximum pixel value, and the hard constraint α is designed to avoid over large regularization weight. Empirically, we set λ^0 as 2×10^{-5} and α as 100. For computational efficiency, we adopt an early stopping mechanism to reduce the number of iterations by monitoring the relative residue. Similar to [1], we define the residue of auxiliary variables as follows:

$$\begin{aligned} \|\Delta \boldsymbol{I}^{(t+1)}\|_2 &= \|\boldsymbol{I}^{(t+1)} - \boldsymbol{I}^{(t)}\|_2, \\ \|\Delta \boldsymbol{U}^{(t+1)}\|_2 &= \|\boldsymbol{U}^{(t+1)} - \boldsymbol{U}^{(t)}\|_2, \\ \|\Delta \boldsymbol{Z}^{(t+1)}\|_2 &= \|\boldsymbol{Z}^{(t+1)} - \boldsymbol{Z}^{(t)}\|_2. \end{aligned} \tag{55}$$

Regarding the relative residue, we define two early stopping conditions. The first one is *Monotonic Decreasing Condition*:

$$\begin{aligned} ||\Delta \boldsymbol{I}^{(t+1)}||_2 &> ||\Delta \boldsymbol{I}^{(t)}||_2 \quad \mathit{and} \\ ||\Delta \boldsymbol{Z}^{(t+1)}||_2 &> ||\Delta \boldsymbol{Z}^{(t)}||_2 \quad \mathit{and} \\ ||\Delta \boldsymbol{U}^{(t+1)}||_2 &> ||\Delta \boldsymbol{U}^{(t)}||_2. \end{aligned} \tag{56}$$

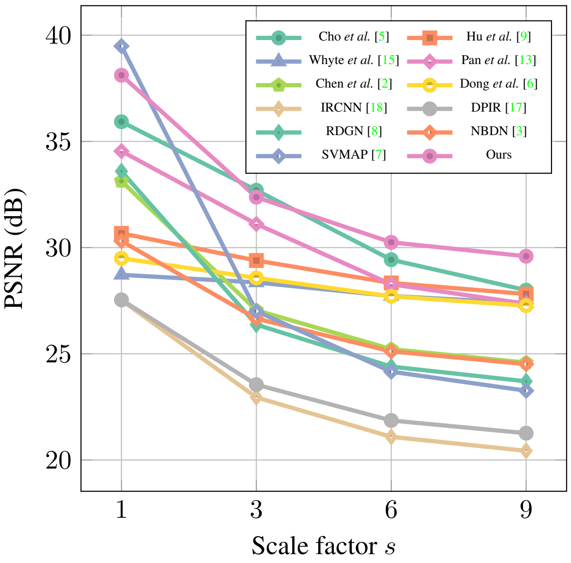
It monitors whether the difference between primal and dual variables decreases monotonically. The second one is *Saturation Awareness Difference Sum Condition*:

$$\frac{1}{\sqrt{n}}||\Delta \boldsymbol{I}^{(t+1)}||_2 + \frac{1}{\sqrt{2n}}||\Delta \boldsymbol{U}^{(t+1)}||_2 + \frac{1}{\sqrt{2n}}||\Delta \boldsymbol{Z}^{(t+1)}||_2 \leq \gamma / \tilde{S}^{(t+1)} \tag{57}$$

and it considers whether the sums of primal and dual residues are smaller than the saturation awareness threshold. For saturated pixels with a higher maximum pixel value, more iterations come with better results, so we set the saturation awareness threshold as $\gamma/\tilde{S}^{(t+1)}$. For pixels with high $\tilde{s}^{(t+1)}$, $\gamma/\tilde{S}^{(t+1)}$ is a relatively tighter threshold, and it comes with more iterations; for pixels with low $\tilde{s}^{(t+1)}$, the constraint of $\gamma/\tilde{S}^{(t+1)}$ is relatively loose, and it comes with fewer iterations. If the residue of primal and dual variables satisfies the above two conditions, the proposed method stops early for computational efficiency.

6. Analysis of the Robustness to Different Scale Factors

To evaluate the robustness of the proposed algorithm to different scale factors, we follow the synthetic scheme of Hu *et al.* [9], which stretches the intensity of the pixels s times beyond a specific value and then clips to the sensor range $[0, 1]$. The source of the synthetic 154 image pairs is from the saturated dataset [9]. The performance of deblurred results is summarized in Fig. 11. In the normal case where the scale factor s equals 1, the proposed algorithm competes with the state-of-the-art learning-based method, SVMAP [7]. However, the performance of SVMAP [7] dramatically declines as the scale factor s increases, and our algorithm still has a competitive performance when the scale factor s is high ($s = 6$ and $s = 9$). That is to say, our algorithm is robust to different scale factors because the saturation awareness mechanism (SAM) adopts proper mask settings for different degrees of saturation.



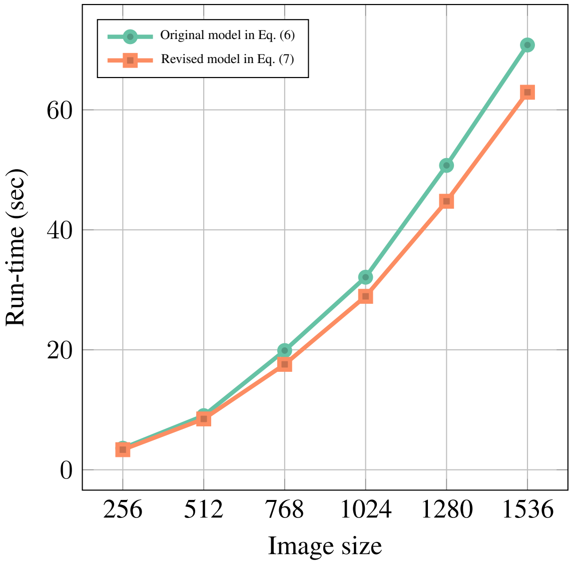


Fig. 11. Analysis of the robustness to different scale factors.

Fig. 12. Analysis of the speed of blur models w.r.t image sizes.

7. Analysis of the Indicator Prior

The indicator prior is used to ensure the pixel values stay positive in each iteration. Without the indicator prior, negative pixel values of $I_{\mathbb{U}}$ can still be feasible solutions to Eq. (8), which will result in erroneous results. Table 4 shows that the PSNR/SSIM values are $29.49/0.9155$ and $\mathbf{29.98/0.9548}$ for the cases without and with the prior, respectively, in the saturated dataset [9]. Since the other two datasets [3, 12] do not activate the *highly-exposed* mode, the indicator function makes no difference to those two datasets.

Table 4. Analysis for the indicator function.

	Saturated	
Prior	PSNR	SSIM
w/o indicator prior	29.49	0.9155
w/ indicator prior	29.98	0.9548

8. Analysis of the Run-time Comparison

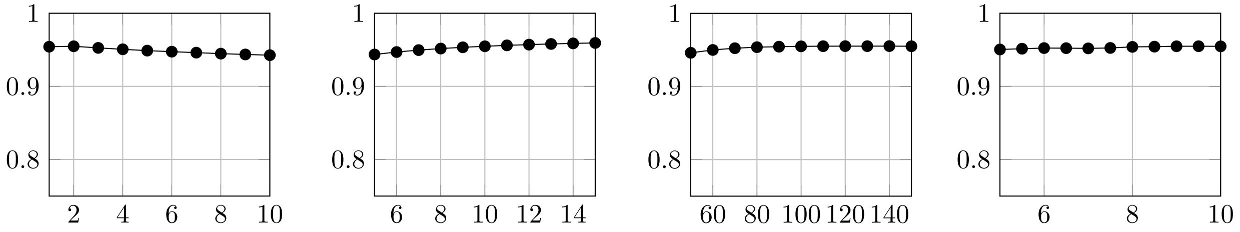
Table 5 summarizes the run-time comparison on a machine with Intel Core i7-9700K CPU @3.60GHz and NVIDIA GeForce RTX 3080 Ti, where the image size is 692x1048x3 pixels. Different from conventional optimization-based methods [2, 5, 9, 13, 15], learning-based methods [6–8, 17, 18] usually have better inference speed. However, the long training time and GPU resources can be issues, not to mention the extra effort when new datasets with different characteristics are added. Although our method has a longer inference time, the proposed handling scheme is robust to saturated pixels with different degrees of saturation. Moreover, our method uses existing DNN priors in our optimization flow. Thus, no extra training process is required for nighttime deblurring.

Table 5. Run-time comparison.

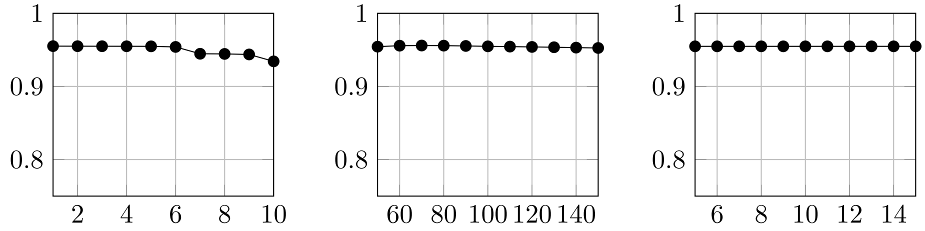
	Cho [5]	Hu [9]	Whyte [15]	Pan [13]	Chen [2]	Dong [6]	IRCNN [18]	DPIR [17]	RDGN [8]	NBDN [3]	SVMAP [7]	Ours
Enable GPU	-	-	-	-	-	-	v	v	v	v	v	v
Run-time (sec)	22.99	15.50	9.44	67.75	17.56	15.27	0.70	10.98	3.99	1.51	1.83	20.26

9. Analysis of the Hyper-parameters

We elaborate on the meaning of all parameters: λ in Eq. (13) controls the denoiser strength, where the larger weight comes with smoother results. ρ in Eq. (11) represents the penalty parameter of the auxiliary variable in the ADMM method. T is the total number of iterations in the ADMM method, and more come with better convergence. ϕ is the hard threshold for the image segment mask in Eq. (3). β in Eq. (51) is the threshold for the *highly-exposed* mode. The hard constraint α in Eq. (17) is designed to avoid an overlarge regularization weight in our adaptive scheme. γ in Eq. (57) is the threshold of the early stopping condition. The effects of the above parameters are evaluated on the saturated dataset [9]. As shown in Fig. 13, the impacts of varying the hyper-parameters within a reasonable range (i.e., similar orders of magnitudes) are subtle to the proposed algorithm.



(a) SSIM with different $\lambda(10^{-5})$ (b) SSIM with different $\rho(10^{-2})$ (c) SSIM with different T (d) SSIM with different $\phi(10^{-1})$



(e) SSIM with different β (f) SSIM with different α (g) SSIM with different γ

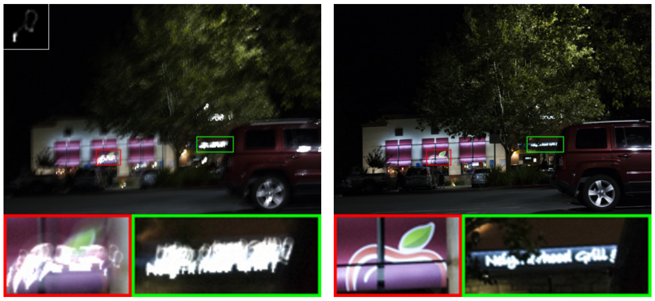
Fig. 13. Sensitivity analysis of hyper-parameters.

10. Analysis of the Speed of Revised Blur Model

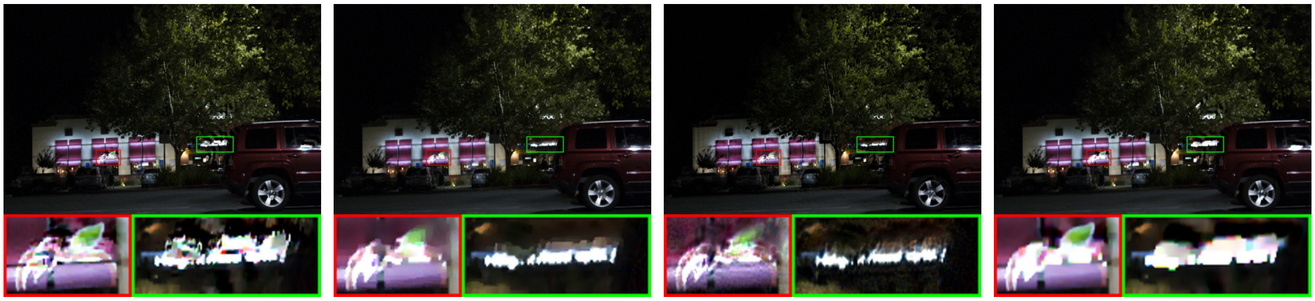
In Section. 3.2, we rewrite the blur model in Eq. (6) to Eq. (7) for computational efficiency. With this modification, a closed-form solution can be obtained. On the other hand, a Hadamard product will be involved when solving the original model in Eq. (6), which requires a slower iterative conjugate-gradient method. We evaluate the run-time on images with respect to image sizes from 256x256x3 to 1536x1536x3. As shown in Fig. 12, the proposed model performs better than the original model regarding the run-time, especially for large image sizes.

11. More Experimental Results on the Benchmark Datasets

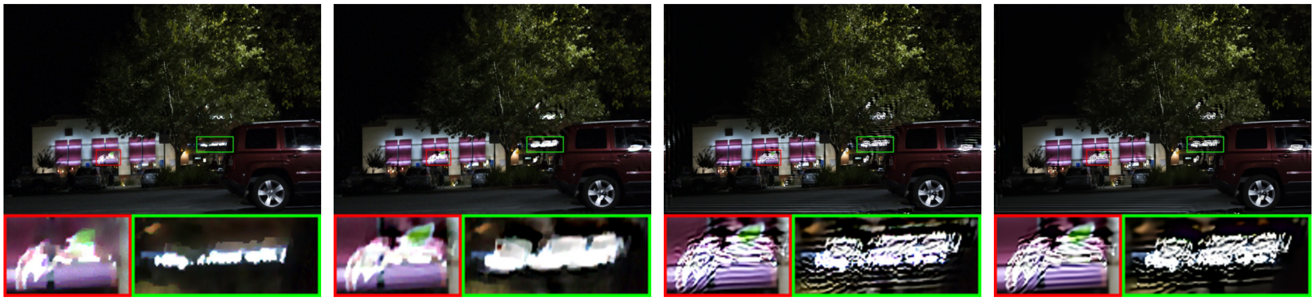
In this section, we visualize more deblurred results with conventional optimization-based methods [2, 5, 9, 13, 15] and state-of-the-art learning-based methods [3, 6–8, 17, 18]. All the experimental results are generated by publicly available code provided by the original authors to give a fair comparison. The deblurred results of the saturated dataset from Hu *et al.* [9] are visualized in Fig. 14 – Fig. 16, and the deblurred results of the low-illumination dataset from Pan *et al.* [12] are visualized in Fig. 17 – Fig. 18. Besides, the deblurred results of the night dataset from Chen *et al.* [3] are visualized in Fig. 19 – Fig. 21, and the deblurred results of real-world blurry images are visualized in Fig. 22 – Fig. 23.



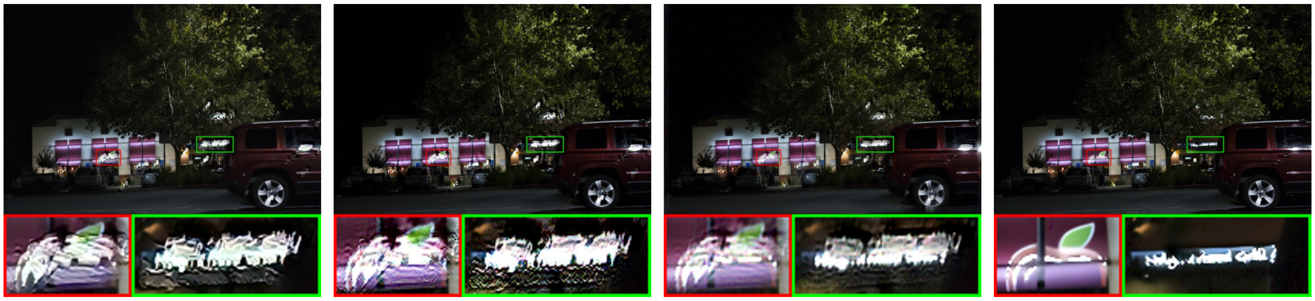
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

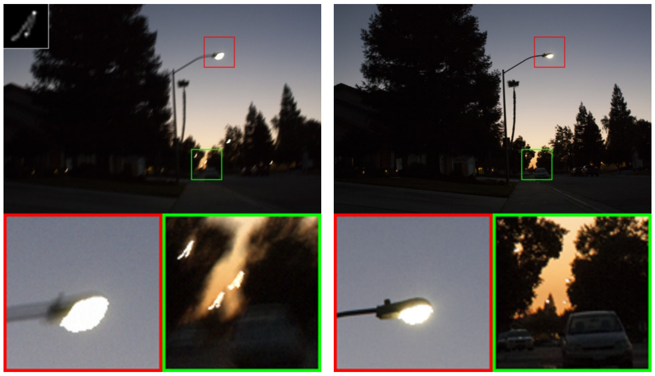


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

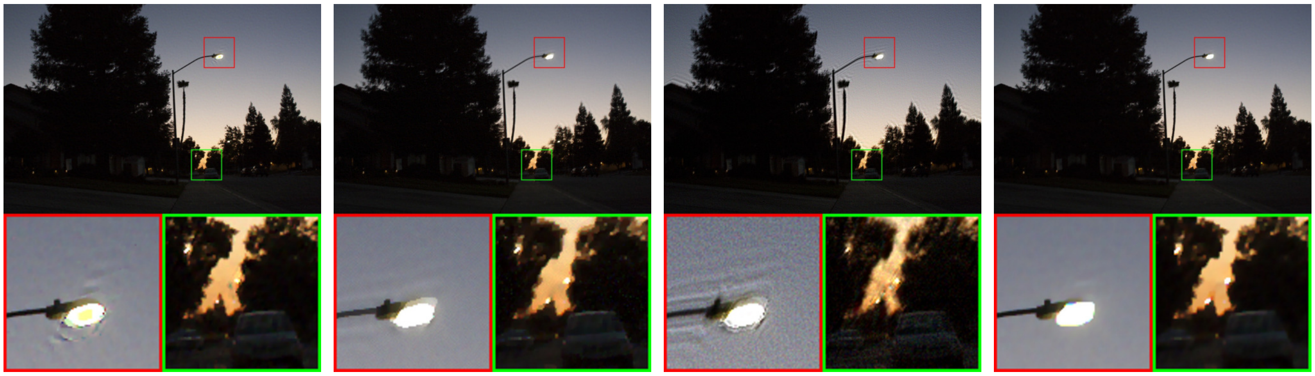


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

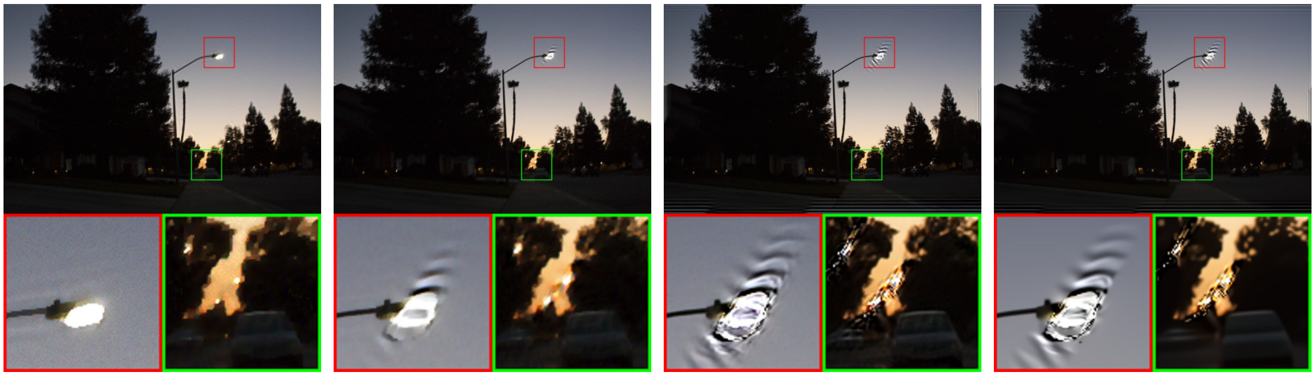
Fig. 14. Comparison of the deblurred results between the proposed method and others.



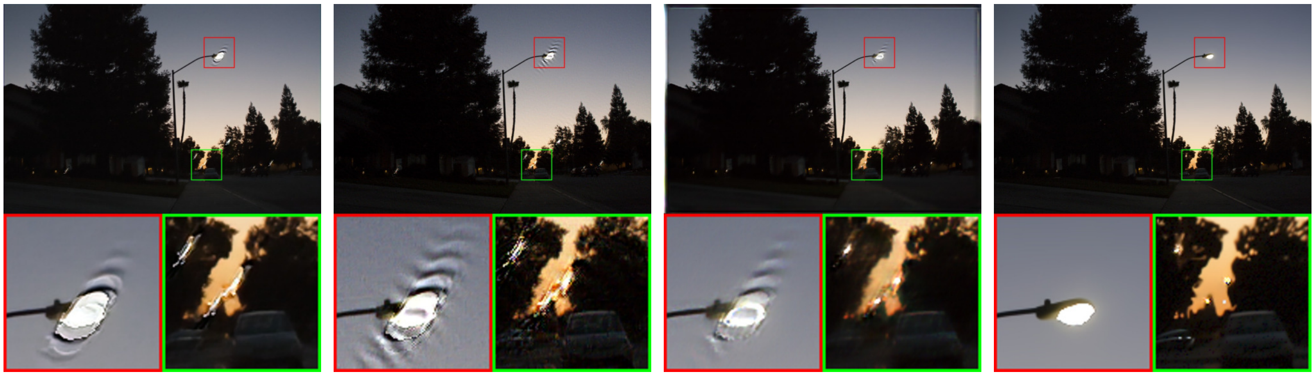
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

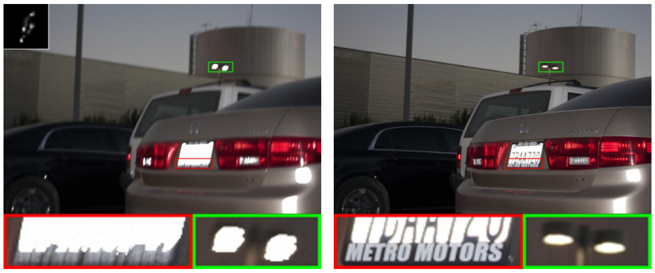


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

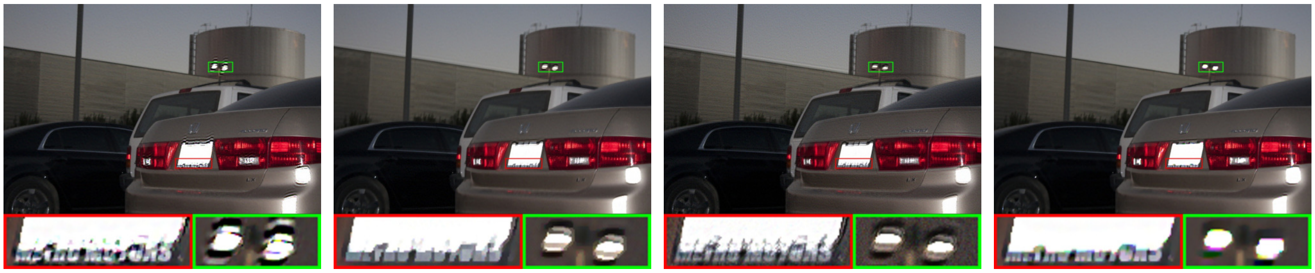


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

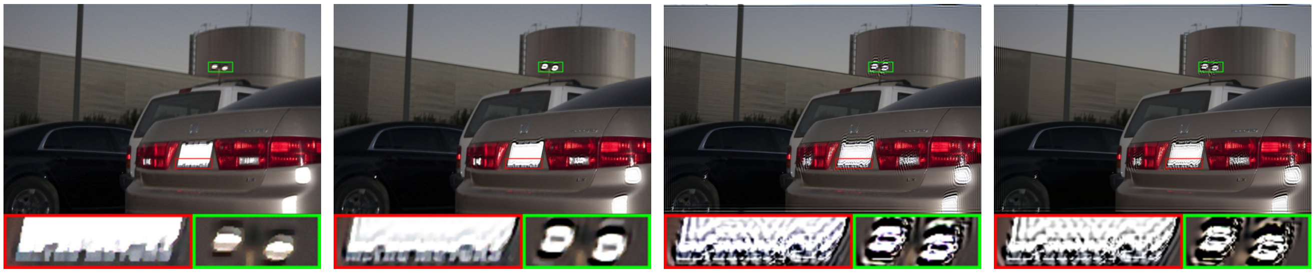
Fig. 15. Comparison of the deblurred results between the proposed method and others.



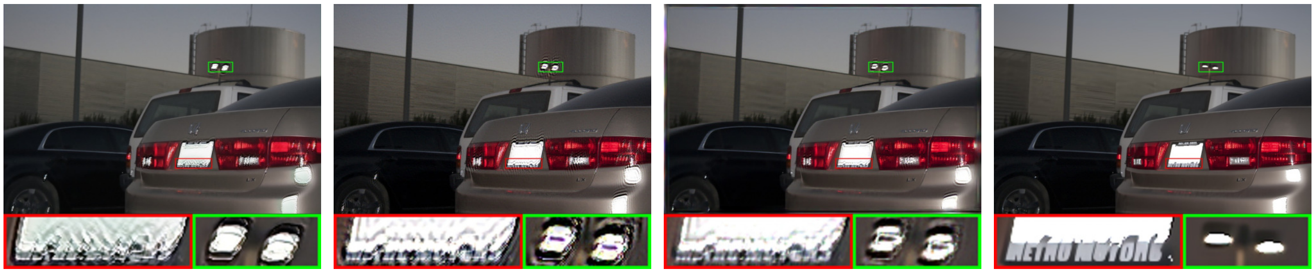
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

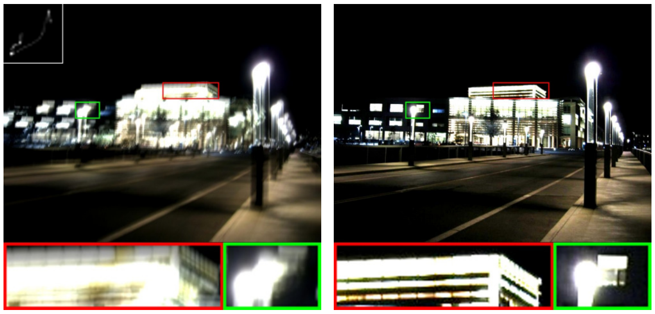


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

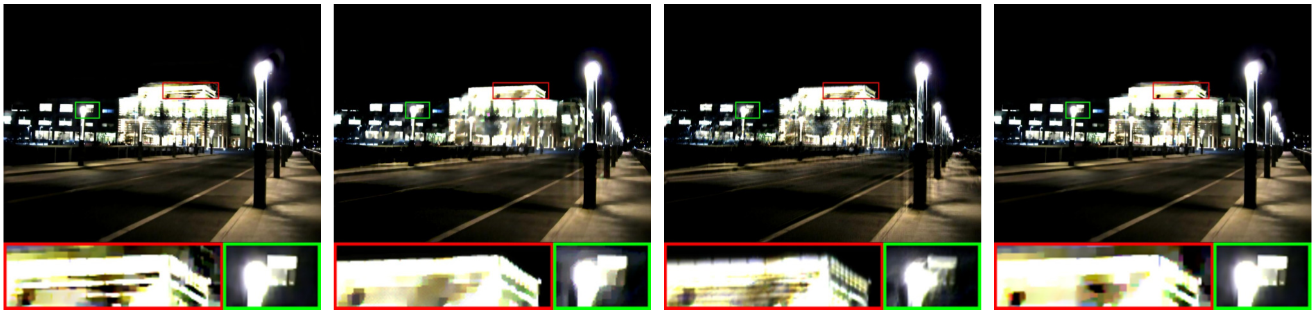


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

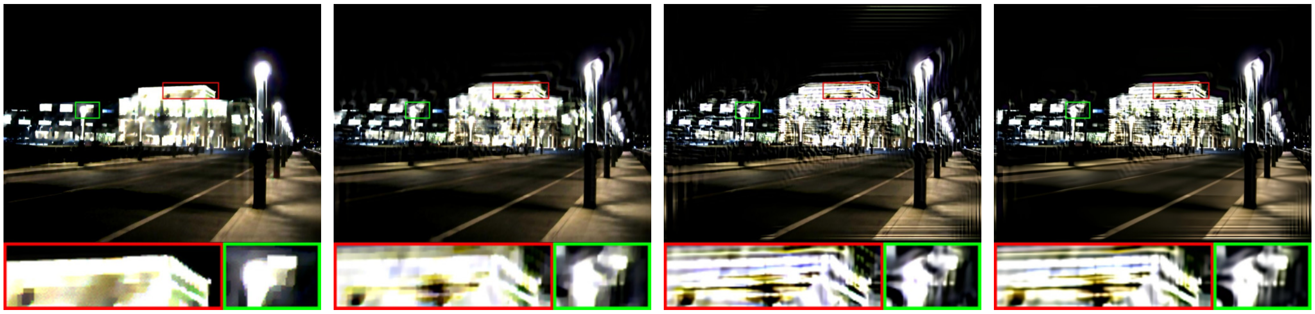
Fig. 16. Comparison of the deblurred results between the proposed method and others.



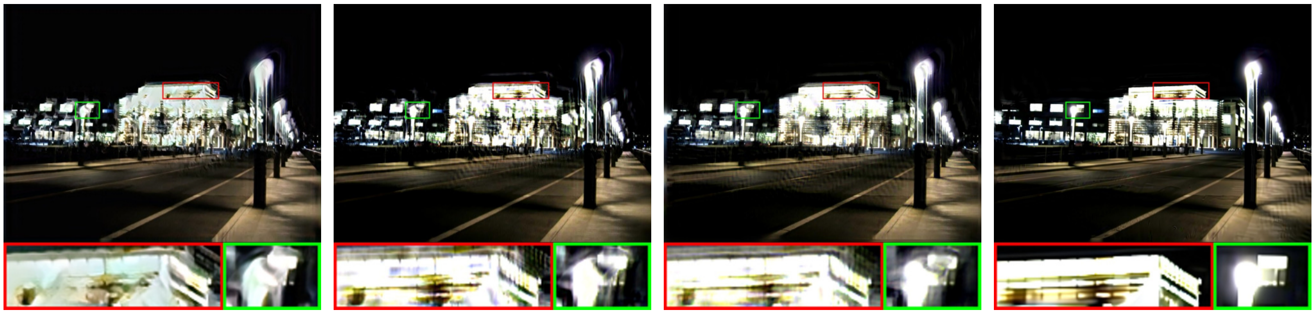
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

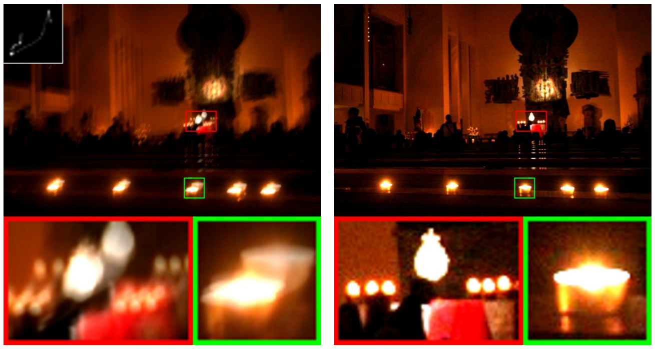


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

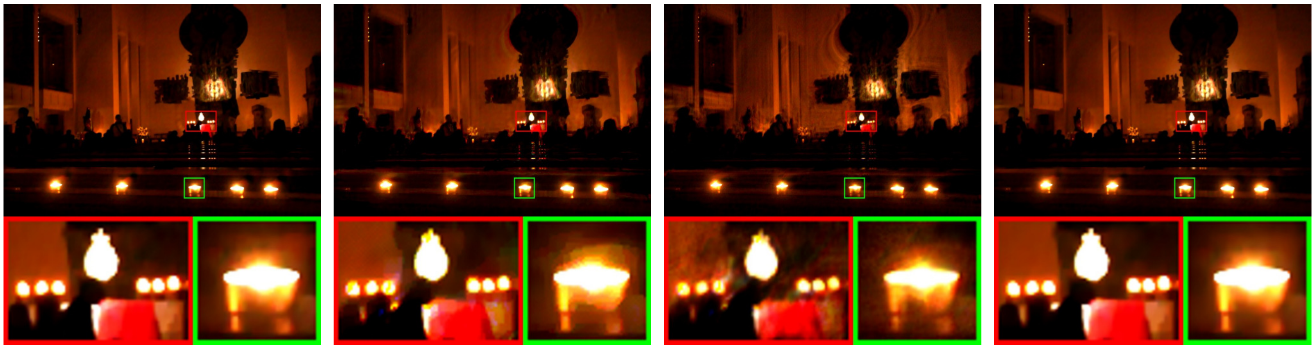


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

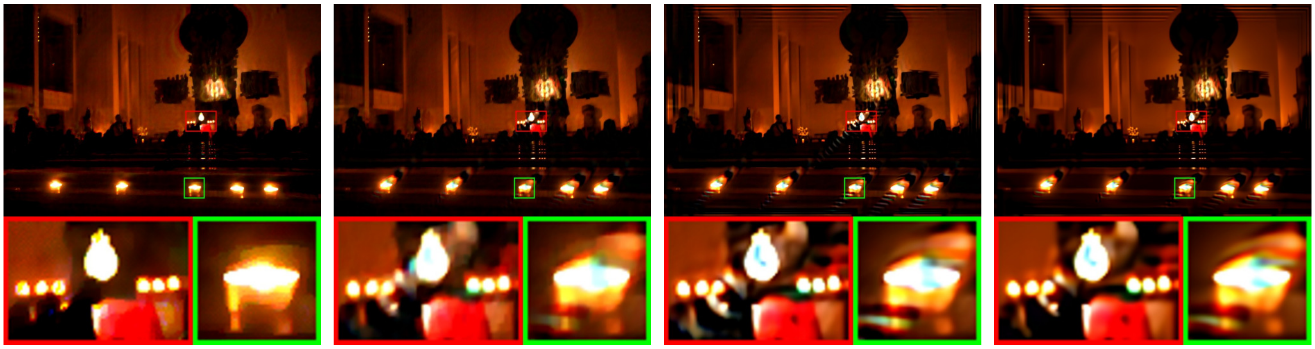
Fig. 17. Comparison of the deblurred results between the proposed method and others.



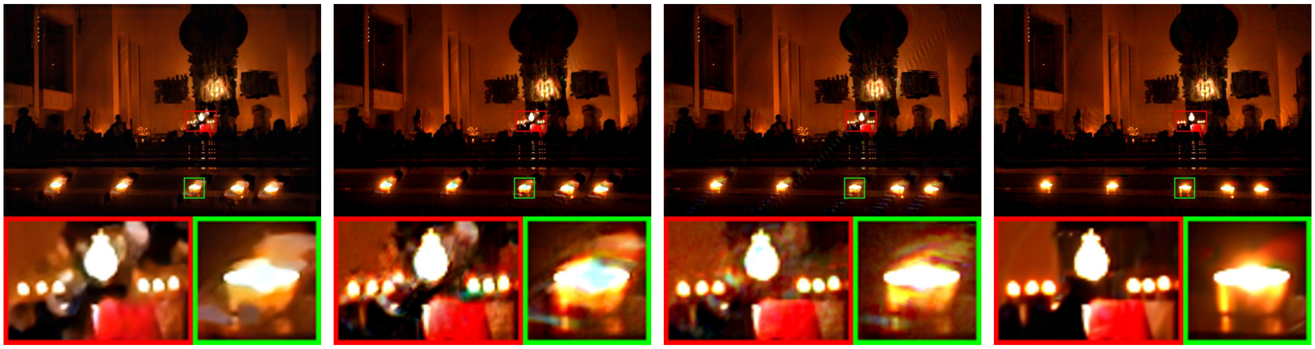
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

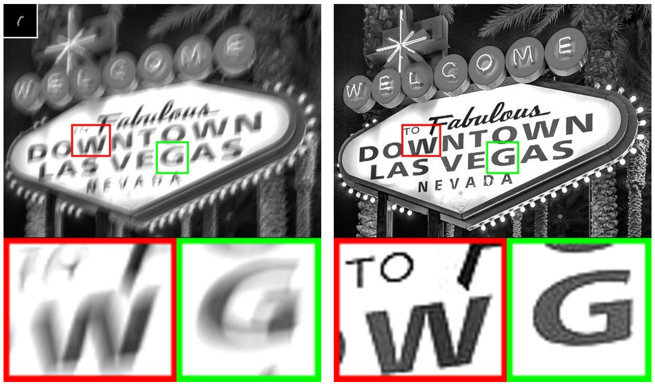


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

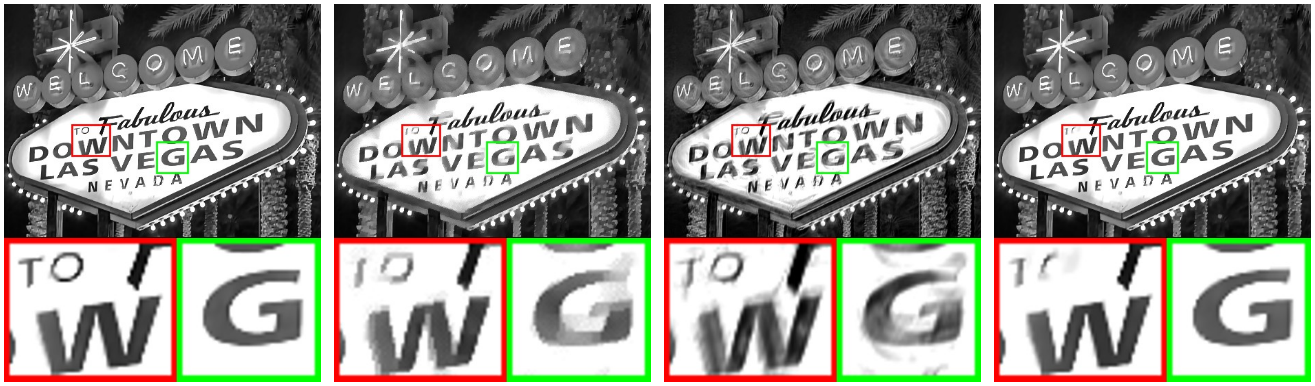


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

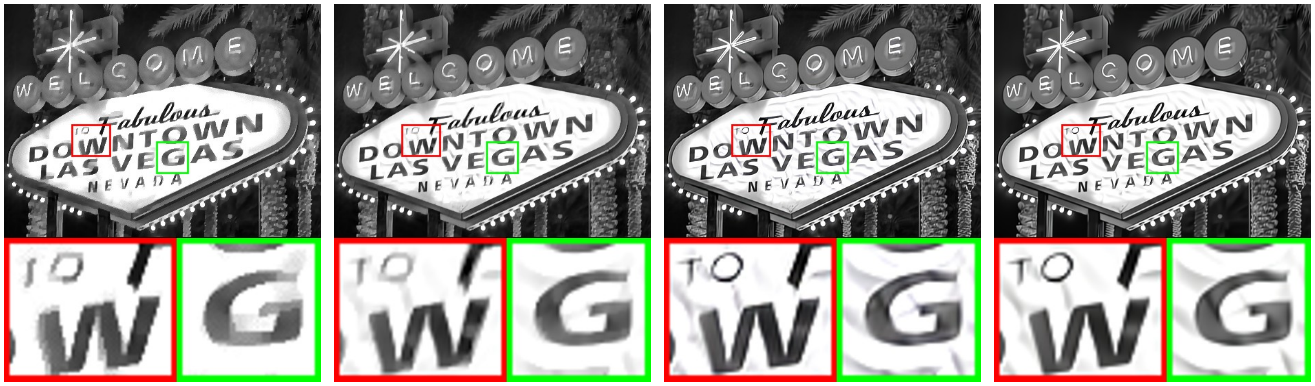
Fig. 18. Comparison of the deblurred results between the proposed method and others.



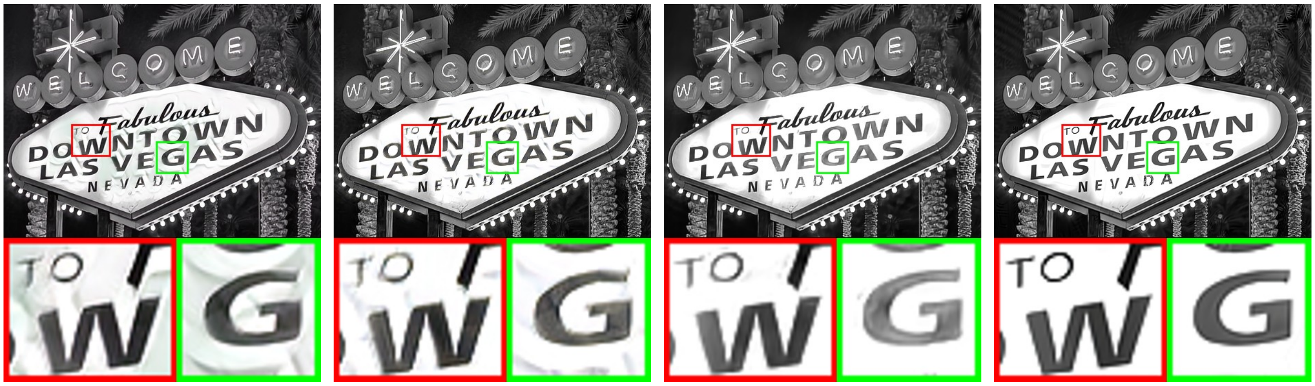
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

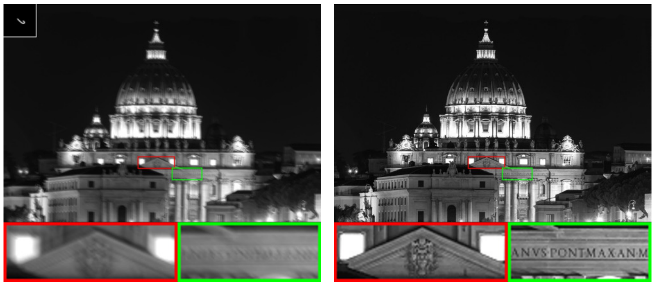


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

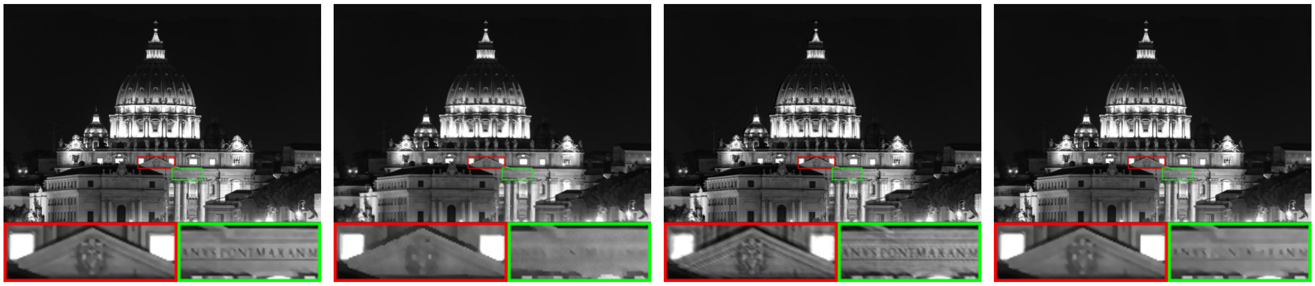


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

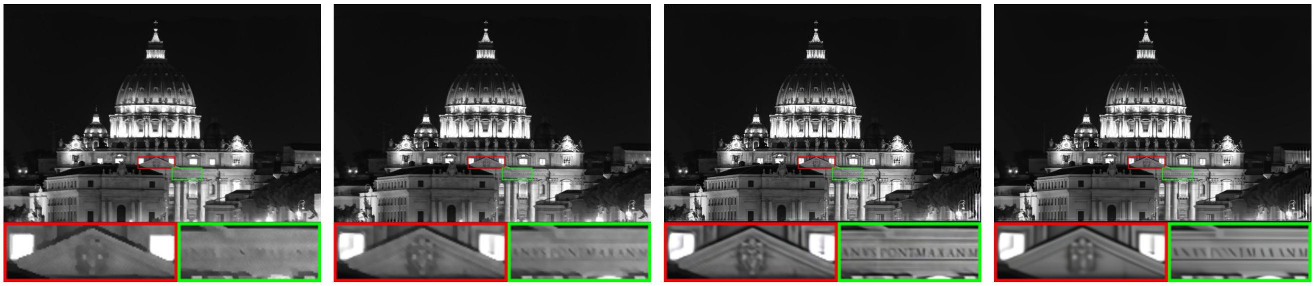
Fig. 19. Comparison of the deblurred results between the proposed method and others.



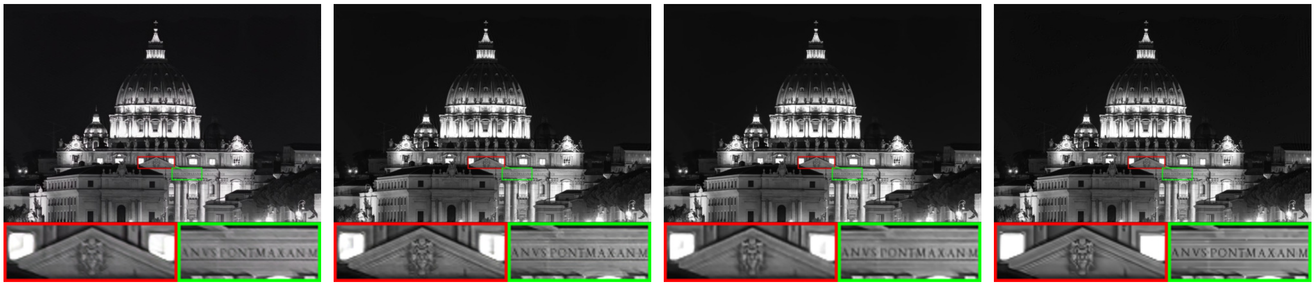
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

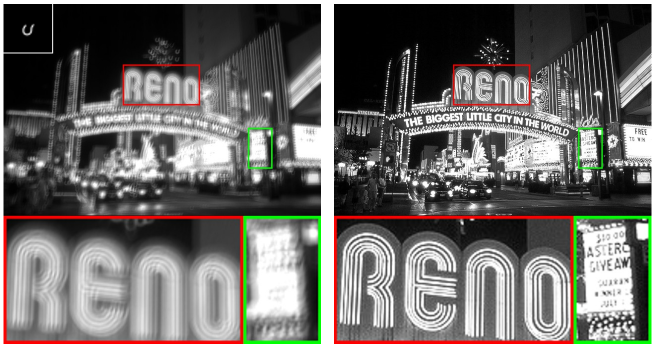


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

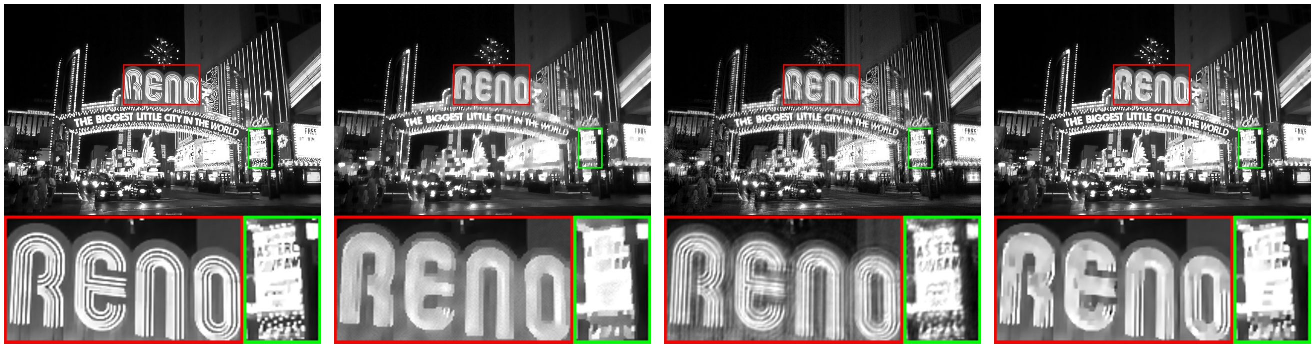


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

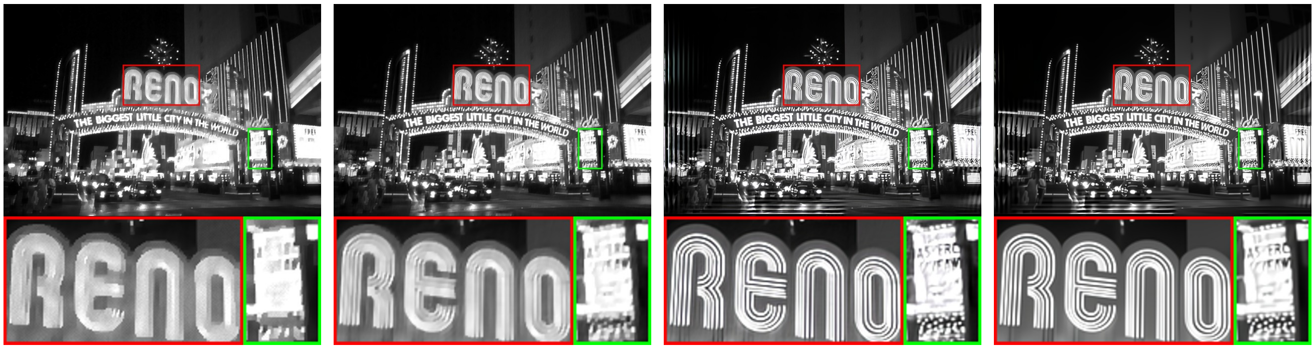
Fig. 20. Comparison of the deblurred results between the proposed method and others.



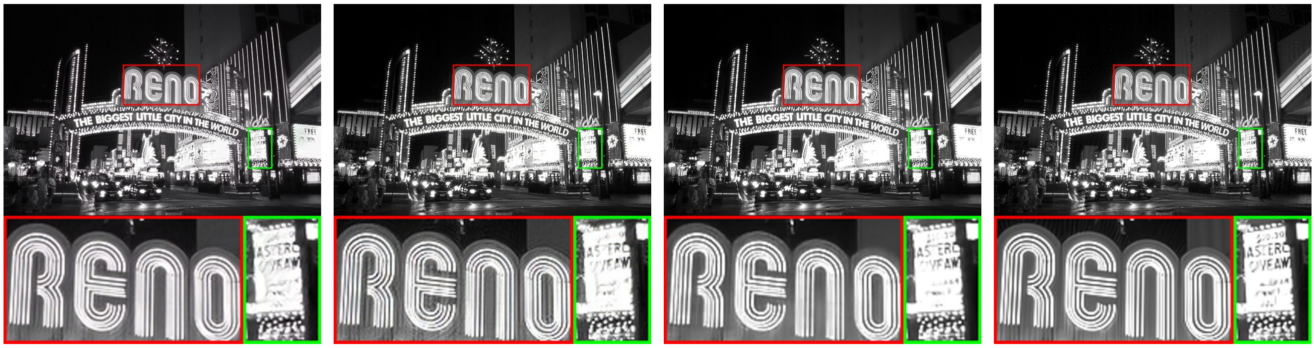
(a) Blurry input (b) Ground Truth



(c) Cho *et al.* [5] (d) Hu *et al.* [9] (e) Whyte *et al.* [15] (f) Pan *et al.* [13]

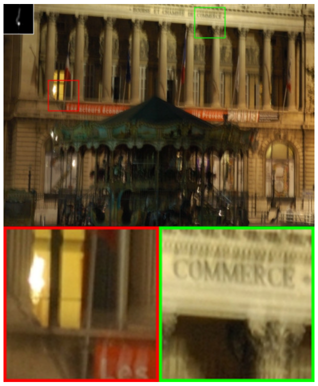


(g) Chen *et al.* [2] (h) Dong *et al.* [6] (i) IRCNN [18] (j) DPIR [17]

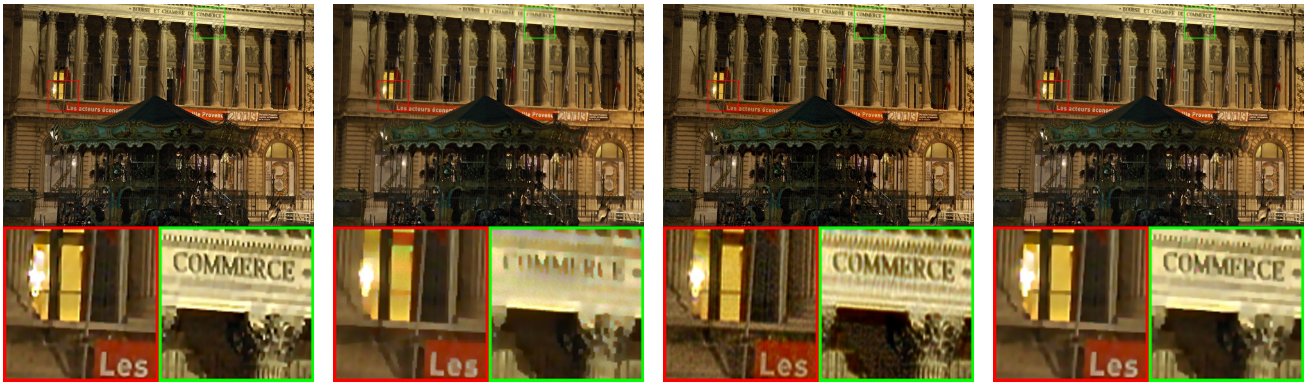


(k) RGDN [8] (l) SVMAP [7] (m) NBDN [3] (n) Ours

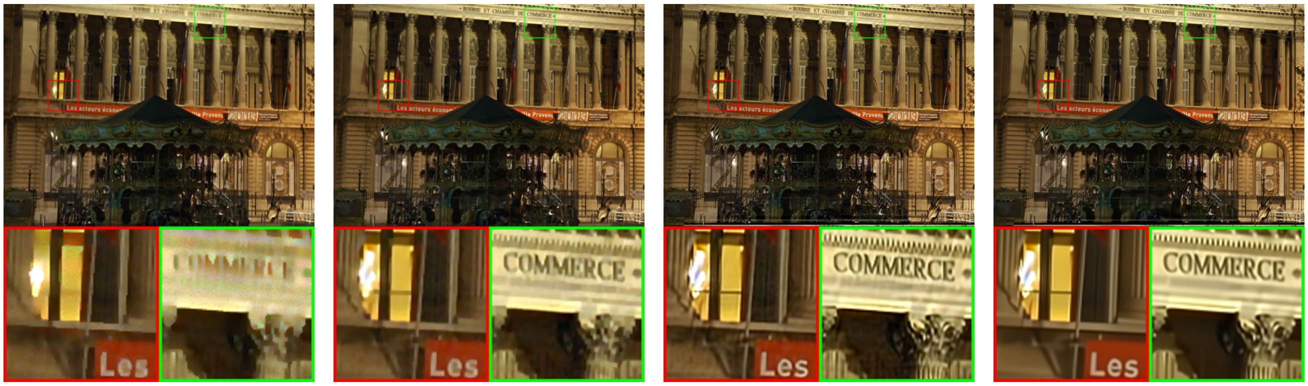
Fig. 21. Comparison of the deblurred results between the proposed method and others.



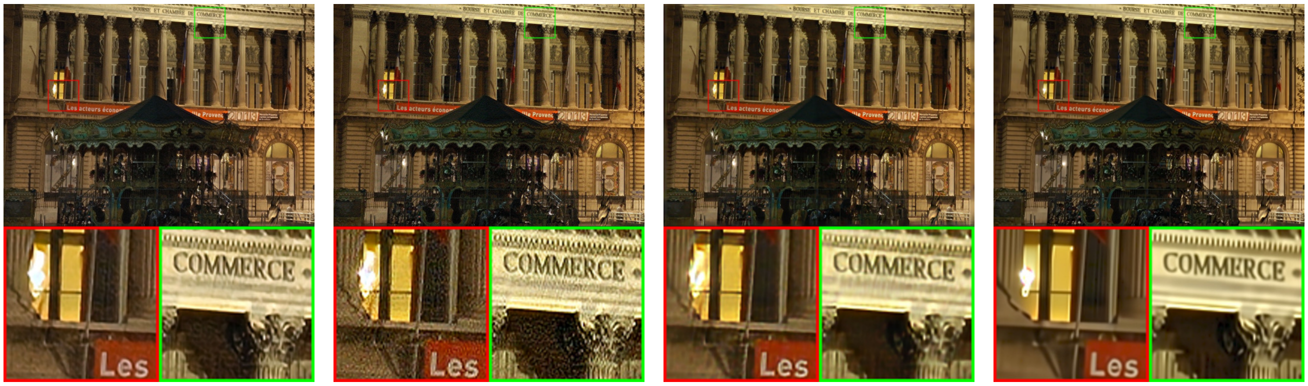
(a) Blurry input



(b) Cho *et al.* [5] (c) Hu *et al.* [9] (d) Whyte *et al.* [15] (e) Pan *et al.* [13]

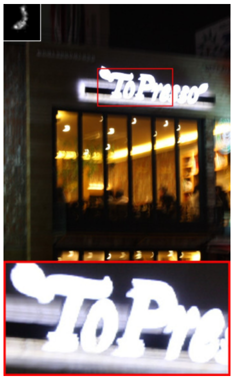


(f) Chen *et al.* [2] (g) Dong *et al.* [6] (h) IRCNN [18] (i) DPIR [17]

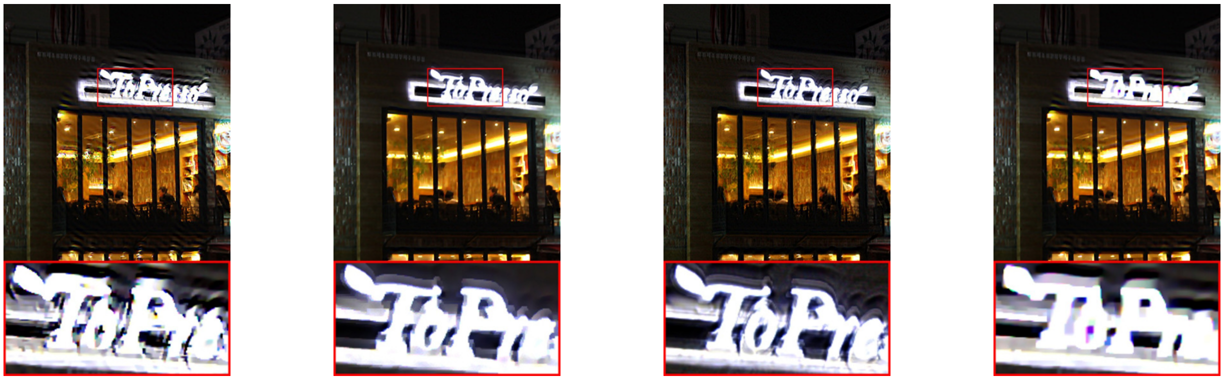


(j) RDGN [8] (k) SVMAP [7] (l) NBDN [3] (m) Ours

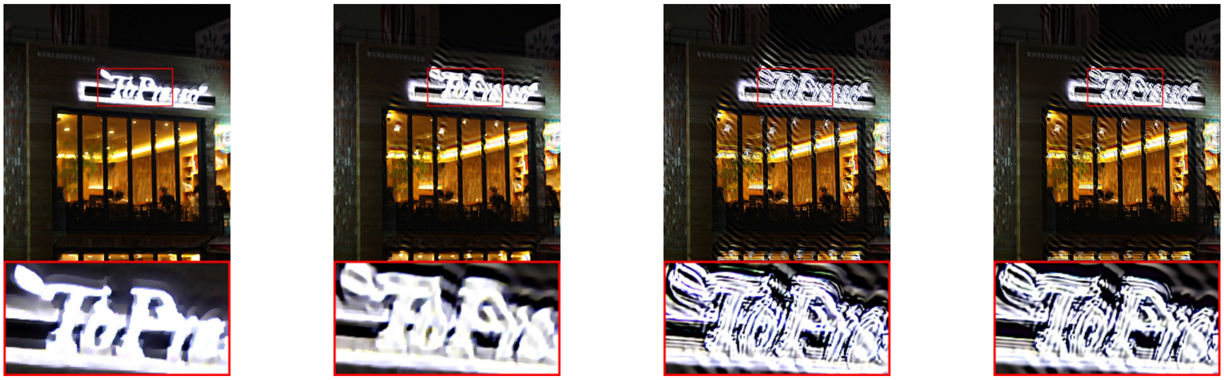
Fig. 22. Comparison of the deblurred results of real world blurred image with saturated pixels from [10] with the kernel estimated by [13].



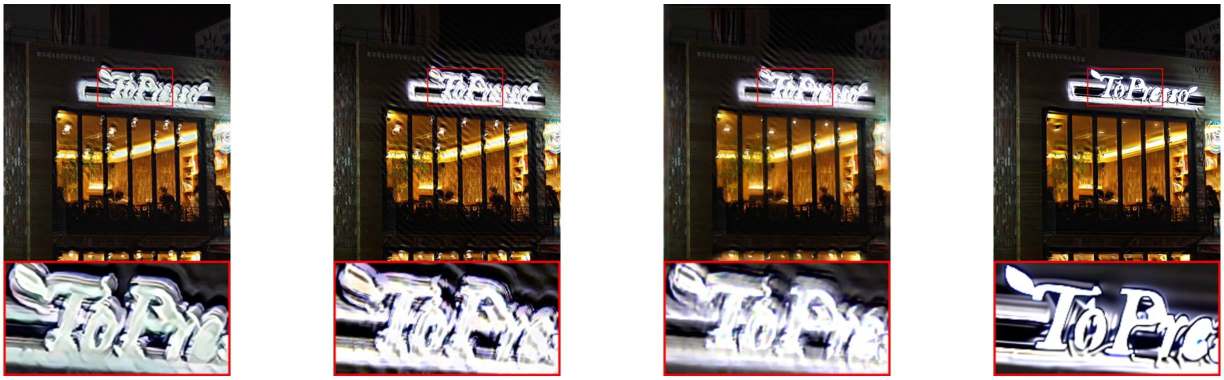
(a) Blurry input



(b) Cho *et al.* [5] (c) Hu *et al.* [9] (d) Whyte *et al.* [15] (e) Pan *et al.* [13]



(f) Chen *et al.* [2] (g) Dong *et al.* [6] (h) IRCNN [18] (i) DPIR [17]



(j) RDGN [8] (k) SVMAP [7] (l) NBDN [3] (m) Ours

Fig. 23. Comparison of the deblurred results of real world blurred image with saturated pixels from [9] with the kernel estimated by [9].

References

[1] Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016. 10

[2] Liang Chen, Jiawei Zhang, Songnan Lin, Faming Fang, and Jimmy S Ren. Blind deblurring for saturated images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6308–6316, 2021. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[3] Liang Chen, Jiawei Zhang, Jinshan Pan, Songnan Lin, Faming Fang, and Jimmy S Ren. Learning a non-blind deblurring network for night blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10542–10550, 2021. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[4] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–8. 2009. 8

[5] Sunghyun Cho, Jue Wang, and Seungyong Lee. Handling outliers in non-blind image deconvolution. In *2011 International Conference on Computer Vision*, pages 495–502. IEEE, 2011. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[6] Jiangxin Dong, Jinshan Pan, Deqing Sun, Zhixun Su, and Ming-Hsuan Yang. Learning data terms for non-blind deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 748–763, 2018. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[7] Jiangxin Dong, Stefan Roth, and Bernt Schiele. Learning spatially-variant map models for non-blind image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4886–4895, 2021. 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[8] Dong Gong, Zhen Zhang, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, and Yanning Zhang. Learning deep gradient descent optimization for image deconvolution. *IEEE Transactions on Neural Networks and Learning Systems*, 31(12):5468–5482, 2020. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[9] Zhe Hu, Sunghyun Cho, Jue Wang, and Ming-Hsuan Yang. Deblurring low-light images with light streaks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3382–3389, 2014. 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[10] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1709, 2016. 21

[11] Renting Liu and Jiaya Jia. Reducing boundary artifacts in image deconvolution. In *2008 15th IEEE International Conference on Image Processing*, pages 505–508. IEEE, 2008. 5

[12] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. l_0-regularized intensity and gradient prior for deblurring text images and beyond. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):342–355, 2016. 11, 12

[13] Jinshan Pan, Zhouchen Lin, Zhixun Su, and Ming-Hsuan Yang. Robust kernel estimation with outliers handling for image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2808, 2016. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[14] Gordon Wetzstein. Ee 367/cs 448i computational imaging and display notes: Noise, denoising, and image reconstruction with noise (lecture 10). 2

[15] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Deblurring shaken and partially saturated images. *International journal of computer vision*, 110(2):185–201, 2014. 1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[16] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *European conference on computer vision*, pages 157–170. Springer, 2010. 5

[17] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[18] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22