LEDNet: Joint Low-light Enhancement and Deblurring in the Dark

Shangchen Zhou, Chongyi Li, and Chen Change Loy

S-Lab, Nanyang Technological University, Singapore
{s200094, chongyi.li, ccloy}@ntu.edu.sg
https://shangchenzhou.com/projects/LEDNet

Abstract. Night photography typically suffers from both low light and blurring issues due to the dim environment and the common use of long exposure. While existing light enhancement and deblurring methods could deal with each problem individually, a cascade of such methods cannot work harmoniously to cope well with joint degradation of visibility and sharpness. Training an end-to-end network is also infeasible as no paired data is available to characterize the coexistence of low light and blurs. We address the problem by introducing a novel data synthesis pipeline that models realistic low-light blurring degradations, especially for blurs in saturated regions, e.g., light streaks, that often appear in the night images. With the pipeline, we present the first large-scale dataset for joint low-light enhancement and deblurring. The dataset, **LOL-Blur**, contains 12,000 low-blur/normal-sharp pairs with diverse darkness and blurs in different scenarios. We further present an effective network, named **LEDNet**, to perform joint low-light enhancement and deblurring. Our network is unique as it is specially designed to consider the synergy between the two inter-connected tasks. Both the proposed dataset and network provide a foundation for this challenging joint task. Extensive experiments demonstrate the effectiveness of our method on both synthetic and real-world datasets.

1 Introduction

When capturing images at night, one would usually use a slow shutter speed (long exposure) to allow more available light to illuminate the image. Even so, the captured dark images may still suffer from low visibility and distorted color induced by insufficient light, which is constrained by minimum shutter speeds that are acceptable for handheld shooting in the dark. Annoyingly, long exposure inevitably causes motion blurs due to camera shake and dynamic scenes. Thus, both low light and motion blurs typically co-exist in in the night images.

Prior methods address the two tasks independently, i.e., low-light enhancement [17,42,9] and image deblurring [11,30,36,48,16,51,5]. These methods made independent assumptions in their specific problem. As a result, a forceful combination cannot solve the joint degradation caused by low light and motion blur. Specifically, existing low-light enhancement methods [42,22] perform intensity

| (a) Input | (b) RUAS | (c) RUAS → MIMO-UNet | (d) MIMO-UNet | (e) LEDNet (Ours) |

Fig. 1. A comparison on the real-world night blurry images shows that existing low-light enhancement and deblurring methods fail in coping with the night blurry images. (a) Input images. (b) Motion blur in the saturated area is enlarged after performing light enhancement using a contemporary method RUAS [22] (indicated by red arrows). (c) Applying the deblurring network MIMO-UNet [7] after light enhancement still fails in blur removal. (d) MIMO-UNet trained on daytime GoPro dataset fails to remove blur in the nighttime images. (e) The proposed LEDNet trained with our LOL-Blur dataset yields satisfactory results through joint low-light enhancement and deblurring.

boosting and denoising, ignoring spatial degradation of motion blurs. Instead, motion blur is even enlarged in saturated regions due to over-exposing after performing light enhancement, as shown in Fig. 1(b). Low-light enhancement methods [42,54] also have the risk of removing informative clues for blur removal due to over-smoothing while denoising. Fig. 1(c) shows that performing deblurring after light enhancement fails the blur removal. As for deblurring, existing methods [36,48,16,7] trained on the datasets that only contain daytime scenes, and thus, cannot be directly applied to the non-trivial night image deblurring. In particular, motion cues in dark regions are poorly visible and perceived due to the low dynamic range, posing a great challenge for these existing deblurring methods. Furthermore, night blurry images contain saturated regions (e.g., light streaks) in which the pixels do not conform to the blur model learned from daytime data [11,5]. As observed in Fig. 1(d), the deblurring network trained on daytime GoPro dataset fails to remove blur in the night images.

The solution to the aforementioned problems is to train a single network that addresses both types of degradations jointly. Clearly, the main obstacle is the availability of such data that come with low-light blurry and normal-light sharp image pairs. The collection is laborious and hard, if not impossible. Existing datasets for low-light enhancement, e.g., LOL [42] and SID [3], gather low-/normal-light pairs by changing exposure time and ISO in two shots. While deblurring datasets, e.g., RealBlur [32], need to capture paired blurry/sharp images under the long and short exposures using a dual-camera system. However, it is challenging to merge these two data collection processes harmoniously to capture paired data for this joint task. Moreover, the existing synthetic deblur-

Sharp image   Previous blur simulation Our blur simulation   Examples in LOL-Blur dataset
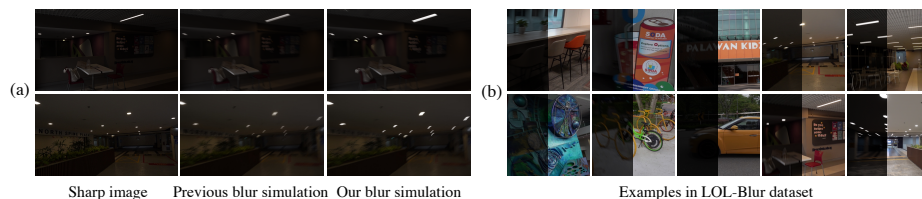
Fig. 2. (a) Comparison on blur simulation. The previous blur simulation [34,30,29], i.e., simple averaging of a sharp sequence, tends to weaken blurs in saturated pixels. By contrast, our simulation generates more realistic saturated blurs that maintain the saturated intensities during blur synthesis. (b) Paired images in the LOL-Blur dataset, containing diverse darkness and blurs (saturated and unsaturated) in the dark.

ring datasets [34,30,29] (e.g., GoPro) cannot simulate blurs of saturated regions for night images due to the lack of sequences that contain saturated scenes and their inappropriate blur simulation by simply averaging. As observed in Fig. 2(a) (middle), their blur simulation method tend to undesirably attenuate blurs of saturated areas, which do not resemble the real ones shown in Fig. 1(a).

This paper makes the first attempt to propose a novel data synthesis pipeline for joint low-light enhancement and deblurring. In particular, we circumvent the difficulty of obtaining dark and blurry images through a heuristic approach for simulating low-light degradation, and a new blur simulation method that pays special attention to model blurs in saturated regions correctly. Figure 2 shows that our blur simulation generates more realistic saturated blurs than the previous ones [34,30,29], maintaining saturated intensities during blur synthesis. The resulting dataset, **LOL-Blur**, contains 12,000 pairs of low-blur/normal-sharp pairs for training and testing. Examples of LOL-Blur are shown in Fig. 2(b).

Apart from the data, we show that it is beneficial to consider both low-light enhancement and deblurring in a single context. In particular, we demonstrate a novel encoder-decoder pipeline, **L**ow-light **E**nhancement and **D**eblurring **Net**work (**LEDNet**), where the encoder is specialized in light enhancement and the decoder in deblurring. The encoder and decoder are linked with adaptive skip connections. This unique structure allows the passing of light enhanced features in the decoder for blur removal in the decoder.

The main contributions: 1) We introduce a novel data synthesis pipeline that models low-light blur degradation realistically, leading to the large-scale and diverse LOL-Blur dataset for joint low-light enhancement and deblurring. 2) We propose a unified network LEDNet with delicate designs to address low-light enhancement and deblurring jointly. 3) We present to aggregate hierarchical global prior that is crucial for stable training and artifacts suppression, as well as the learnable non-linear layer helps brighten dark areas without overexposing other regions. 4) Extensive experiments show that our method achieves superior results to prior arts on both synthetic and real-world datasets.

2   Related Work

Image Deblurring. Many CNN-based methods have been proposed for dynamic scene deblurring [30,36,49,48,16,51,7]. Most early studies [35,8] employ networks to estimate the motion blur kernels followed by non-blind methods. Owing to the emergence of training datasets for deblurring tasks [34,30,29,33,58,19,32], end-to-end kernel-free networks become the dominant methods. To obtain a large receptive field, some networks [30,36,7] adopt a multi-scale strategy to handle large blurs Similarly, some multi-patch deblurring networks [51,47,10] employ the hierarchical structures without down-sampling. GAN-based deblurring methods [15,16] have been proposed to generate more details. To deal with spatially-varying blurs, Zhang *et al.* [49] propose spatially variant RNNs to remove blur via estimating RNN weights. Zhou *et al.* [57] propose the filter adaptive convolutional (FAC) layer to handle non-uniform blurs dynamically. In our paper, we built a filter adaptive skip connection between encoder and decoder using FAC layers.

Optimization-based approaches are proposed for low-light image deblurring [11,4,5,6]. Hu *et al.* [11] suggest the use of light streaks to estimate blur kernel. However, their method heavily relies on light streaks and tends to fail when the light sources are not available or too large beyond pre-designed blur kernel size. Chen *et al.* [5,6] process saturated regions specially and ensure smaller contributions of these pixels in optimization. Their results show few artifacts around saturated regions. While effective, all these methods are time-consuming, thus limiting their applicability.

Low-light Enhancement. Deep networks have become the mainstream in low-light enhancement (LLE) [17]. The first CNN model LL-Net [23] employs an autoencoder to learn denoising and light enhancement simultaneously. Inspired by the Retinex theory, several LLE networks [42,54,38,44,22] are proposed. They commonly split a low-light input into reflectance and illumination maps, then adjust the illumination map to enhance the intensity. Most methods integrate a denoising module on the reflectance map for suppressing noise in the enhanced results. For example, Zheng *et al.* [56] propose an unfolding total variation network to estimate noise level for LLE. While the joint task of LLE and deblurring has not been investigated yet in the literature.

To improve the generalization capability, some unsupervised methods are proposed. EnlightenGAN [13] is an attention-based U-Net trained using adversarial loss. Zero-DCE [9] and Zero-DCE++ [18] formulate light enhancement as a task of image-specific curve estimation. Their training adopts several manually-defined losses on supervision of exposure or color, without limitation of paired or unpaired training data. Thus, Zero-DCE can be easily extended to generic lighting adjustments. Notably, due to the pixel-wise curve adjustment formulation, it can be used for spatially-varying light adjustment. In our data synthesis pipeline, we train an exposure conditioned Zero-DCE to darken images for low-light simulation. Given random low exposure degrees, we can generate low-light images of diverse darkness.
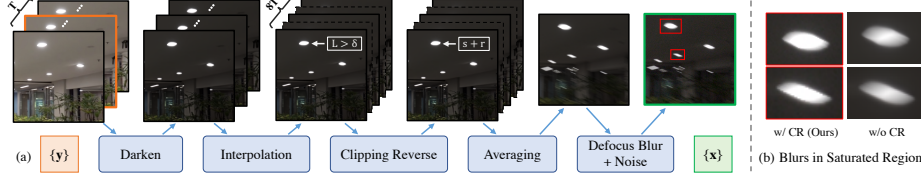
Fig. 3. (a) An overview of our data synthesis pipeline. (b) Comparisons on two blur simulations in the saturated regions. With the proposed Clipping Reverse (CR), we can generate realistic blurs with sharp boundaries in saturated regions, which better resembling real cases that are caused by the large light ratio in night photography.

3   LOL-Blur Dataset

Efforts have been made to collect real-world paired data for low-light enhancement [42,21,3,2,12,37] or image deblurring [32], but not both settings at the same time. The lack of such data is not surprising as (1) Paired images of low-light enhancement datasets and image deblurring datasets are commonly collected by different camera shot settings, and (2) The collection of both kinds of data is susceptible to geometric and photometric misalignment due to camera shake or dynamic environment during data acquisition.

In this work, inspired by the big success of data synthesis in the real-world super-resolution tasks [40,50,20,1], we introduce a synthesis pipeline that models low-light blur degradation jointly, hence allowing us to generate a large-scale dataset (LOL-Blur). We acquire a total of 170 videos for training and 30 videos for testing, each of which has 60 frames, amounting to 12,000 paired data.

3.1   Existing Synthesis Methods and Limitations

Low-light Simulation. Prior works [23,25] use Gamma correction to simulate low-light images, defined by a nonlinearity power-law expression $I_{low} = \alpha I_{in}^{\gamma}, (\gamma > 1)$, where constant α is usually set to 1. This synthetic process tends to introduce large color deviation with noticeable warm tones [17], in contrast, our simulation method EC-Zero-DCE (refer to Sec. 3.2) produce more natural and realistic low-light images. A comparison is provided in the supplementary.
Blur Simulation. A standard synthesis pipeline of blurry data [34,30,29,58,19] is to average successive frames on high frame-rate sequences for approximating the blur model [30]. The process can be expressed as:

$$B = g\left(\tfrac{1}{T} \sum_{i=0}^{T-1} S[i]\right) = g\left(\tfrac{1}{T} \sum_{i=0}^{T-1} g^{-1}\left(\hat{S[i]}\right)\right), \tag{1}$$

where $g(\cdot)$ is CRF function (Gamma curve with $\gamma = 2.2$) that maps latent signal $S[i]$ into observed sRGB images $\hat{S[i]}$. This process can be used to generate blurry-sharp pairs for daytime scenes, assuming $\hat{S[i]} = g\left(S[i]\right)$. This blur model,

however, is usually not accurate to the regions of saturated pixels that often appear in dark blurry images. This is because the saturated intensities in latent signal $S[i]$ are clipped to the maximum value (255) when $S[i]$ is saved as an sRGB image $\hat{S}[i]$, due to the limited dynamic range of sRGB images, i.e., $\hat{S}[i] = Clip(g(S[i]))$. This clipping function damages the exceeding value of saturated regions, thus making the blur model of Eq. (1) improper for these regions [5]. Our simulation pipeline resolves this issue by recovering the clipped intensities in saturated regions (refer to Sec. 3.2), generating more realistic light blurs. As a visual comparison are shown in Fig. 2(a).

3.2   Data Generation Pipeline

The overview of our data generation pipeline is shown in Fig. 3. We use a Sony RX10 IV camera to record 200 high frame-rate videos at 250 fps. With the video sequences, we first downsize each frame to a resolution of 1120×640 to reduce noises. We then apply VBM4D [27] for further denoising and obtain the clean sequences. In our method, we take 7 or 9 frames as a sequence clip, as shown in Fig. 3. The mid-frame (with orange bounding box) among the sharp frames is treated as the ground truth image. Then, we process the following steps to generate low-light and blurred images.

Darkening with Conditional Zero-DCE. To simulate the degradation of low light, we reformulate the Zero-DCE [9] into an Exposure-Conditioned variant, EC-Zero-DCE. Contrary to Zero-DCE that is designed for improving the brightness of an image, EC-Zero-DCE simulates low light with controllable darkness levels, via implementing a reversed curve adjustment. Specifically, we modify the exposure control loss by replacing the fixed exposure value with a random parameter that represents darkness while other losses are kept in the same settings as Zero-DCE. Given different exposure levels, EC-Zero-DCE can generate realistic low-light images with diverse darkness. Note that EC-Zero-DCE performs pixel-wise and spatially-varying light adjustment, rather than uniform light degradation. We provide the luminance adjustment map in the supplementary to support this statement.

Frame Interpolation. To avoid discontinuous blurs in the synthetic blurry images, we increase the frame rate to 2000 fps using a high-quality frame interpolation network [31].

Clipping Reverse for Saturated Region. To restore the clipped intensity in saturated regions that were ignored by the previous blur simulation (i.e., Eq. (1)), a simple yet effective way is by adding a random supplementary value $r \sim \mathcal{U}(20, 100)$ to RGB channels in these regions. We first define the saturated regions where lightness channel $L > \delta$ in the Lab color space, the threshold δ is empirically set to 98 in our pipeline, where $L \in [0, 100]$. Then, we reformulate the blur model as a more general form for both saturated and unsaturated regions, as shown in Eq. (2):

$$B = g\left(\tfrac{1}{T} \sum_{i=0}^{T-1} Clip^{-1}\left(g^{-1}\left(\hat{S}[i]\right)\right)\right), \tag{2}$$

where $Clip^{-1}(s) = s + r$ if s in the saturated regions, otherwise $Clip^{-1}(s) = s$. Fig. 2(a) and Fig. 3(b) shows our blur simulation using clipping reverse (w/ CR) generates more realistic saturated blurs than the GoPro blur simulation (w/o CR) that is commonly used in previous datasets. Moreover, the modified blur simulation (w/ CR) indeed helps networks handle well on both unsaturated and saturated blurs, as indicated by the comparison in Fig. 8.

Frame Averaging. Next, we average 56 (7×8) or 72 (9×8) successive frames of 2000 fps videos to produce virtual blurry videos at around 24 fps.

Adding Defocus Blur and Noise. To generate more realistic low-light blurry images, our pipeline also considers defocus blurs by applying generalized Gaussian filters [40]. We also add realistic noises into low-blur images generated by CycleISP [45]. Both defocus blur and noise are added in a random fashion.

Discussion. Our dataset offers realism in low-light blur degradation and consists of 200 common dynamic dark scenarios (indoor and outdoor) with diverse darkness and motion blurs, as shown in Fig. 2(b). Compared to previous synthetic deblurring datasets (such as GoPro [30] and REDS [29]) that only contain daytime scenes and lacks the saturated regions, our dataset contains a total of 55 sequences with various sources of artificial lights that often appear in the night photography. Hence, our simulated data sufficiently covers hard cases with blurs in saturated areas, e.g., light streaks, which are indispensable for our joint task. Experimental results demonstrate that the networks trained using our dataset generalizes well on real-world dark blurred images.

4   LEDNet

We treat the joint task of low-light enhancement (LLE) and deblurring as a non-blind image restoration problem. The low-light blurry images $\{x\}$ contain the mixed degradations of visibility and texture. The two type degradations are spatially-varying due to local lighting conditions and dynamic scene blurs. To solve this issue, we specially design a network, LEDNet, to map low-light blurry images $\{x\}$ to its corresponding normal-light sharp images $\{y\}$. As shown in Fig. 4, LEDNet is built upon an encoder-decoder architecture with filter adaptive skip connections to solve this joint spatially-varying task.

4.1   Low-light Enhancement Encoder

The encoder (LE-Encoder) is designed for Low-light Enhancement with the supervision of intermediate enhancement loss (see Sec. 4.4). It consists of three scale blocks, each of which contains one Residual Block, one Residual Downsampling Block [46], a Pyramid Pooling Module (PPM) [55], and a Curve Non-Linear Unit (CurveNLU), as shown in Fig. 4. To facilitate intermediate supervision, we output an enhanced image by one convolution layer at the smallest scale. Our design gears LE-Encoder to embed the input image x into the feature space of normal-light images, allowing the subsequent decoder (D-Decoder) to pay more attention to the deblurring task.
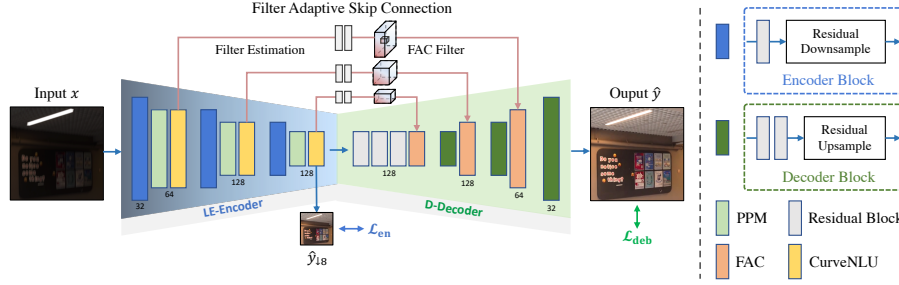
Fig. 4. An illustration of the proposed LEDNet. It contains an Encoder for Light Enhancement, LE-Encoder, and a Decoder for Deblurring, D-Decoder. They are connected by three Filter Adaptive Skip Connections. The PPM and CurveNLU layers are inserted in LE-Encoder, making light enhancement more stable and powerful. LEDNet applies spatially-adaptive transformation to D-Decoder using filters generated by FASC from enhanced features. CurveNLU and FASC enable LEDNet to perform spatially-varying feature transformation for both intensity enhancement and blur removal.

Pyramid Pooling Module. The outputs of typical light enhancement networks are often prone to local artifacts, especially when the networks are fed with high-resolution inputs. We found that the problem can be significantly remedied by injecting global contextual prior into the networks. To achieve this goal, we introduce PPM into our LE-Encoder. The PPM extracts hierarchical global prior using multi-scale regional pooling layers and aggregates them in the last convolution layer. We adopt the original design of PPM that has four mean pooling branches with bin sizes of $1, 2, 3, 6$, respectively. The is the first time PPM is used in a low-light enhancement network. We show that it is crucial for suppressing artifacts that may be caused by the co-existence of other degradations of blur and noise (refer to a comparison shown in Fig. 9).

Curve Non-Linear Unit. Local lighting such as light sources are often observed in the night environment. A global operator tends to over- or under-enhance these local regions. To solve this problem, Zero-DCE [9] applies pixel-wise curve parameters to the input image iteratively for light enhancement.

Inspired by Zero-DCE, we propose a learnable non-linear activation function, namely CurveNLU. The CurveNLU is designed for feature transformation using the estimated curve parameters, as shown in Fig. 5. Similar to Zero-DCE, we formulate the high-order curve in an iterative function:

$$\mathcal{C}_n(\mathrm{p}) = \begin{cases} A_1 F(\mathrm{p})(1 - F(\mathrm{p})) + F(\mathrm{p}), & n = 1 \\ A_{n-1}(\mathrm{p})\mathcal{C}_{n-1}(\mathrm{p})(1 - \mathcal{C}_{n-1}(\mathrm{p})) + \mathcal{C}_{n-1}(\mathrm{p}), & n > 1 \end{cases} \tag{3}$$

where p denotes position coordinates of features, and A_{n-1} is the pixel-wise curve parameter for the n-*th* order of the estimated curve. Given an input feature $F \in \mathbb{R}^{H \times W \times C}$, Curve Estimation module estimates curve parameters $A \in \mathbb{R}^{H \times W \times n}$ that represent an $n+1$ order curve for different positions. Feature transformation is then achieved by Eq. 3 using the estimated curve parame-
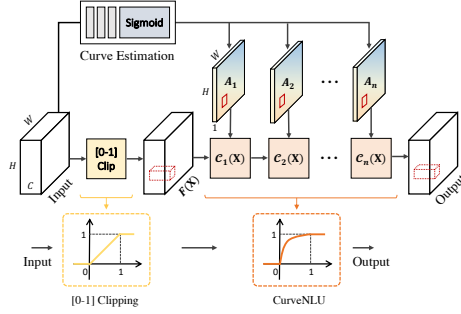
Fig. 5. An illustration of Curve Non-Linear Unit. This layer can be seen as a learnable non-linear activation function between 0 and 1. Based on Eq. 3, the learned function always follows concave down increasing curves to increase feature intensities.

ters. Different from Zero-DCE that uses different curves for RGB channels, our CurveNLU applies the same curve to different channels in the feature domain. Note that the parameters A lay in $[0, 1]$, ensuring that CurveNLU always learns concave down increasing curves to increase the features of dark areas without overexposing other regions. To meet such a requirement, the input feature F of CurveNLU is needed to be clipped to the range of $[0, 1]$ at the beginning. The Curve Estimation module consists of three convolution layers followed by a Sigmoid function. We set the iteration number n to 3 in our experiments.

4.2   Deblurring Decoder

With the enhanced features from LE-Encoder, Deblurring Decoder (D-Decoder) is able to concentrate more on deblurring. It contains three convolutional blocks, each of which has two Residual Blocks, one Residual Upsampling Block [46], and a FAC Layer [57] that is used to bridge the LE-Encoder and the D-Decoder.

4.3   Filter Adaptive Skip Connection

Both low-light enhancement and deblurring in our task are spatially varying problems. Deblurring in the dynamic scenes is challenging due to its spatially variant blurs caused by object motion and depth variations. Though CurveNLU applies pixel-wise adjustment in the LE-Encoder, it is not enough for the deblurring task that usually needs dynamic spatial kernels to handle motion blurs. Filter Adaptive Convolutional (FAC) layer [57] has been proposed to apply dynamic convolution filters for each element in features. Built on the FAC layers, we design a Filter Adaptive Skip Connection (FASC) to solve the deblurring problem by exploiting the enhanced information from LE-Encoder. As shown in Fig. 4, given the enhanced features $E \in \mathbb{R}^{H \times W \times C}$ at different scales, FASC estimates the corresponding filter $K \in \mathbb{R}^{H \times W \times Cd^2}$ via three 3×3 convolution layers and a 1×1 convolution layer to expand the feature dimension. The filter K is then used by FAC layers to transform the features $D \in \mathbb{R}^{H \times W \times C}$ in D-Decoder. For each element of feature D , FAC applies a convolution operator using the corresponding $d \times d$ kernel from the filter K to obtain refined features. We set the kernel size d to 5 at the three scales, following the setting in Zhou *et al.* [57].

4.4   Loss Function

Low-light Enhancement Losses. To provide intermediate supervision, we employ L1 loss and perceptual loss at $\times 8$ downsampled scale. Specifically, we predict the image $\hat{y}_{\downarrow 8}$ for the smallest scale of LE-Encoder, and then restrict it using scale-corresponding ground truth $y_{\downarrow 8}$, shown as Eq. (4):

$$\mathcal{L}_{en} = \|\hat{y}_{\downarrow 8} - y_{\downarrow 8}\|_1 + \lambda_{per} \|\phi(\hat{y}_{\downarrow 8}) - \phi(y_{\downarrow 8})\|_1, \tag{4}$$

where $\phi(\cdot)$ represents the pretrained VGG19 network. We adopt multi-scale feature maps from layer $\{conv1, \cdots, conv4\}$ following the widely-used setting [41]. Due to downsampling space, the enhancement loss $\mathcal{L}_{en}$ mainly supervises the exposure of intermediate output.

Deblurring Losses. We use the L1 loss and perceptual loss as our deblurring loss $\mathcal{L}_{deb}$, defined as follows:

$$\mathcal{L}_{deb} = \|\hat{y} - y\|_1 + \lambda_{per} \|\phi(\hat{y}) - \phi(y)\|_1. \tag{5}$$

The overall loss function is:

$$\mathcal{L} = \lambda_{en}\mathcal{L}_{en} + \lambda_{deb}\mathcal{L}_{deb}. \tag{6}$$

We set the loss weights of λ_{per}, λ_{en}, and λ_{deb} to 0.01, 0.8, and 1, respectively.

5   Experiments

Dataset and Experimental Settings. We train our network LEDNet and other baselines on LOL-Blur dataset. The 170 sequences (10,200 pairs) are used for training and 30 sequences (1,800 pairs) for test. We randomly crop 256×256 patches for training. The mini-batch size is set to 8. We train our network using Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.99$ for a total of 500k iterations. The initial learning rate is set to 10^{-4} and updated with cosine annealing strategy [24].

5.1   Evaluation on LOL-Blur Dataset

We quantitatively and qualitatively evaluate the proposed LEDNet on our LOL-Blur Dataset. Since the joint task is newly-defined in this paper, there is no method available to make a comparison directly. We carefully choose and combine existing representative low-light enhancement and deblurring methods, providing three types of baselines for comparisons. Specifically, the baseline methods lay on following three categories:

1. Enhancement $\rightarrow$ Deblurring. We choose the recent representative light enhancement networks Zero-DCE [9] and RUAS [22] followed by a state-of-the-art deblurring network MIMO-UNet [7].

2. Deblurring $\rightarrow$ Enhancement. For deblurring, we include a recent optimization-based method [5] particularly designed for saturated image deblurring, a GAN-based network DeblurGAN-v2 [16] trained on RealBlur dataset,

Table 1. Quantitative evaluation on our LOL-Blur dataset. PSNR/SSIM↑: the higher, the better; LPIPS↓: the lower, the better. The symbol '†' indicates that we use DeblurGAN-v2 trained on RealBlur dataset, and '*' indicates the network is retrained on our LOL-Blur dataset. The runtime and parameters are expressed in seconds and millions. All runtimes are evaluated on a 720p image using a GPU V100.

Methods	Enhancement → Deblurring		Deblurring → Enhancement			Training on LOL-Blur					
	Zero-DCE [9] → MIMO [7]	RUAS [22] → MIMO [7]	Chen [5] → Zero-DCE [9]	DeblurGAN-v2† [16] → Zero-DCE [9]	MIMO [7] → Zero-DCE [9]	KinD++* [53]	DRBN* [43]	DeblurGAN-v2* [16]	DMPHN* [48]	MIMO* [7]	Ours
PSNR↑	17.68	17.81	17.02	18.33	17.52	21.26	21.78	22.30	22.20	22.41	**25.74**
SSIM↑	0.542	0.569	0.502	0.589	0.57	0.753	0.768	0.745	0.817	0.835	**0.850**
LPIPS↓	0.510	0.523	0.516	0.476	0.498	0.359	0.325	0.356	0.301	0.262	**0.224**
Runtime (s)	-	-	-	-	-	0.06	0.11	0.13	0.26	0.16	0.12
Params (M)	-	-	-	-	-	1.2	0.6	60.9	5.4	6.8	7.4
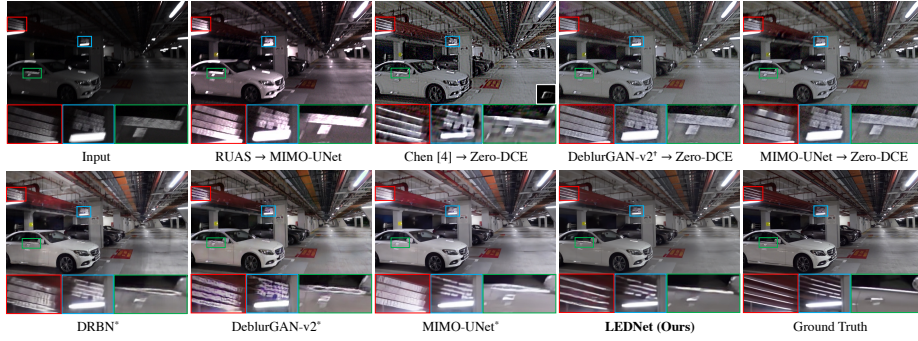
Fig. 6. Visual comparisons on the LOL-Blur dataset. The proposed method generates much sharper images with visually pleasing results. (**Zoom in for best view**)
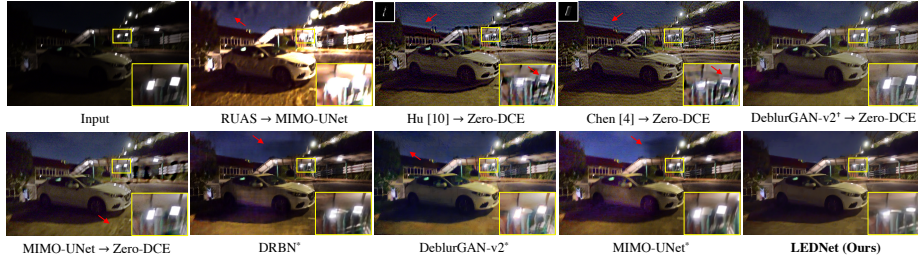
and a state-of-the-art deblurring network MIMO-UNet [7] trained on GoPro dataset. Since RUAS tends to produce overexposed results in the saturated regions that may cover up previous deblurring results, we employ Zero-DCE for light enhancement in this type of baseline.

3. End-to-end training on LOL-Blur dataset. We retrain some state-of-the-art baselines on our dataset using their released code. They include two light enhancement networks KinD++ [53] and DRBN [43], and three deblurring networks of DeblurGAN-v2 [16], DMPHN [48], and MIMO-UNet [7].

Evaluation Metrics. We employ the PSNR and SSIM metrics for evaluation on the synthetic LOL-Blur dataset. To evaluate the perceptual quality of restored images, we also report the perceptual metric LPIPS [52] for references.

Quantitative Evaluations. Table 1 shows quantitative results on our LOL-Blur dataset. The proposed LEDNet performs favorably against other baseline methods. Notably, the better performance at a similar runtime cost and model size of other networks. The results suggest LEDNet is effective and particularly well-suited for this task due to the specially designed network and losses.

Qualitative Evaluations. Fig. 6 compares the proposed LEDNet with baseline methods on LOL-Blur dataset. All compared methods produce unpleasing results

| Input | RUAS → MIMO-UNet | Hu [10] → Zero-DCE | Chen [4] → Zero-DCE | DeblurGAN-v2† → Zero-DCE |

| MIMO-UNet → Zero-DCE | DRBN* | DeblurGAN-v2* | MIMO-UNet* | **LEDNet (Ours)** |

Fig. 7. Visual comparison on a real-world night blurred image. Our method achieves the best perceptual quality with more stable light enhancement and better deblurring performance, while other methods still leave large blurs in saturated regions and suffer from noticeable artifacts, as indicated by red arrows. (**Zoom in for best view**)

and suffer from serious blur artifacts, especially in saturated regions. In contrast, LEDNet generates perceptually pleasant results with sharper textures.

5.2   Evaluation on Real Data

We also collected a real test dataset, named **Real-LOL-Blur**, that contains 240 captured low-light blurry images in the wild and 160 night blurry images from the RealBlur dataset [32].

Evaluation Metrics. As ground-truths are not available for real test images, We employ the recent image quality accessment method: MUSIQ [14], and two wildely-used ones: NRQM [26] and NIQE [28] as our perceptual metrics. We choose MUSIQ model trained on KonIQ-10k dataset, it focuses more on color contrast and sharpness assessment, which is more suitable for our task.

Quantitative Evaluations. As shown in Table 2, the proposed LEDNet achieves the highest MUSIQ score, indicating that our results are perceptually best in terms of color contrast and sharpness. LEDNet also obtains the best NRQM and NIQE scores, showing our results have the best image qualities that are well in line with human perception.

Qualitative Evaluations. Fig. 7 presents a visual comparison on a real-world night blurry image. The methods of Hu *et al.* [11] and Chen *et al.* [5] are particularly designed for saturated image deblurring, however, their cascading baselines still suffer from noticeable artifacts in the presence of large saturated regions. Besides, the end-to-end baseline networks trained on our LOL-Blur dataset are also less effective given the real-world inputs, as their architecture are not specially designed to handle this task. As shown in Fig. 7, their results usually suffer from undesired severe artifacts (red arrows) and blurs (yellow boxes) in their results. Overall, the proposed LEDNet shows the best visual quality, with fewer artifacts and blurs. The better performance is attributed to the CurveNLU and FASC, which enable LEDNet to perform spatially-varying feature transformation for both intensity enhancement and blur removal. The comparisons on real
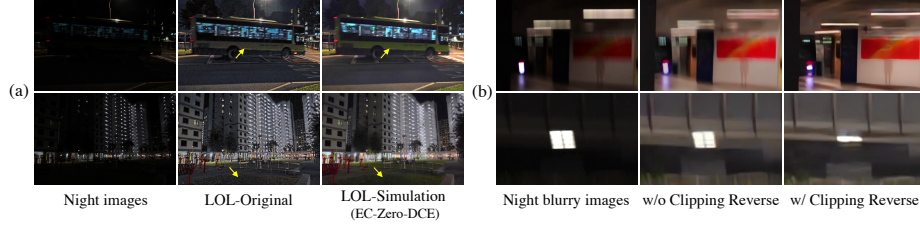
Fig. 8. Ablation study on data synthesis pipeline. (a) Results comparison on different training datasets: original LOL dataset (LOL-Original) and our simulated LOL dataset (LOL-Simulation). The network trained on LOL-Simulation generates more natural results with less noise and color distortion. (b) Results comparison on different data synthesis pipelines. Applying Clipping Reverse in training data generation enables the network to be robust to handle blur in saturated regions.

images strongly suggest the effectiveness of our dataset and network. Notably, benefiting from the noise simulation via CycleISP in the training dataset, our model handles real-world noises well.

Table 2. Evaluation on Real-LOL-Blur.

	RUAS $\rightarrow$ MIMO	MIMO $\rightarrow$ Zero-DCE	KinD++*	DRBN*	DMPHN*	MIMO*	Ours
MUSIQ↑	34.39	28.36	31.74	31.27	35.08	35.37	**39.11**
NRQM↑	3.322	3.697	3.854	4.019	4.470	5.140	**5.643**
NIQE↓	6.812	6.892	7.299	7.129	5.910	4.851	**4.764**

Table 3. Ablation study results of variant networks on LOL-Blur.

	(a) w/o PPM	(b) w/o CurveNLU	(c) Concat	(d) w/o $\mathcal{L}_{en}$	(e) Ours
PSNR↑	21.85	25.20	25.31	24.05	**25.74**
SSIM↑	0.781	0.823	0.826	0.784	**0.850**

5.3   Ablation Study

In this subsection, we present an ablation study to demonstrate the effectiveness of the key steps in data synthesis pipeline and the main modules in LEDNet.

Low-light Simulation using EC-Zero-DCE. To demonstrate the effectiveness of the proposed low-light simulation, we construct a new LOL-simulation dataset by applying our EC-Zero-DC to darken the normal-light images in LOL dataset [42], thus we obtain low-/normal-light paired images for training. We retrain the network KinD++ [53] using the LOL-simulation dataset for comparison with the official model that was trained on the original LOL dataset. Fig. 8(a) shows that our simulated method enables the network to generate more natural results with less noise and color distortion (indicated by the yellow arrows).

Clipping Reverse (CR). Fig. 3(b) shows that CR helps generate more realistic blurs in saturated regions. Fig. 8(b) provides a visual comparison on real-world blurry image, it suggests that applying CR in training data generation helps the network to generalize better in blur removal around saturated regions.

| Input | w/o PPM | w/ PPM |

| Input | Feature Enhancment Rate |

Fig. 9. Result comparison of variant networks: without PPM and with PPM.

Fig. 10. CurveNLU enhancement rate F_{out}/F_{in} of different channels.

Effectiveness of PPM. The PPM layer provides crucial global prior for stable training and artifacts suppression in low-light enhancement. In Table 3(a), The variant LEDNet without Pyramid Pooling Module (w/o PPM) significantly degrades the network performance. Besides, the network removing PPM suffers from noticeable artifacts in the enhanced images, as shown in Fig. 9.

Effectiveness of CurveNLU. Fig. 10 shows the feature enhancement rate F_{in}/F_{out} of input F_{in} and output F_{out} of CurveNLU. As observed, feature adjustment in CurveNLU is spatially adaptive to different regions in the image. The merit of CurveNLU is also validated in Table 3.

Effectiveness of FASC Connections. Comparing variant LEDNet (c) and (e) in Table 3, the one with FASC connection achieves better performance compared to simple connection based on concatenation. This is because the saturated and unsaturated areas in the night scenes follow different blur models. The task in this paper poses more requirements of spatially-varying operations.

Effectiveness of Enhancement Loss. The enhancement loss $\mathcal{L}_{en}$ is necessary in our method. Removing it from training harm the performance as shown in Table 3(d). It is because this intermediate loss helps decompose our joint task into low-light enhancement and deblurring, which makes it easier to optimize.

6   Conclusion

We have presented a novel data synthesis pipeline to model realistic low-light blurring. Based on the pipeline, we built a large-scale and diverse paired dataset (LOL-Blur) for learning and benchmarking the new joint task of low-light enhancement and deblurring. We have also proposed a simple yet effective model, LEDNet, which performs illumination enhancement and blur removal in a single forward pass. We showed that PPM is beneficial and introduced CurveNLU to make the learned network more stable and robust. We further described FASC for better deblurring. Our dataset and network offer a foundation for further exploration for low-light enhancement and deblurring in the dark.

References

1. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Investigating tradeoffs in real-world video super-resolution. In: CVPR (2022)
2. Chen, C., Chen, Q., Do, M.N., Koltun, V.: Seeing motion in the dark. In: ICCV (2019)
3. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: CVPR (2018)
4. Chen, L., Fang, F., Zhang, J., Liu, J., Zhang, G.: Oid: Outlier identifying and discarding in blind image deblurring. In: ECCV (2020)
5. Chen, L., Zhang, J., Lin, S., Fang, F., Ren, J.S.: Blind deblurring for saturated images. In: CVPR (2021)
6. Chen, L., Zhang, J., Pan, J., Lin, S., Fang, F., Ren, J.S.: Learning a non-blind deblurring network for night blurry images. In: CVPR (2021)
7. Cho, S.J., Ji, S.W., Hong, J.P., Jung, S.W., Ko, S.J.: Rethinking coarse-to-fine approach in single image deblurring. In: CVPR (2021)
8. Gong, D., Yang, J., Liu, L., Zhang, Y., Reid, I., Shen, C., Van Den Hengel, A., Shi, Q.: From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In: CVPR (2017)
9. Guo, C., Li, C., Guo, J., Loy, C.C., Hou, J., Kwong, S., Cong, R.: Zero-reference deep curve estimation for low-light image enhancement. In: CVPR (2020)
10. Hu, X., Ren, W., Yu, K., Zhang, K., Cao, X., Liu, W., Menze, B.: Pyramid architecture search for real-time image deblurring. In: ICCV (2021)
11. Hu, Z., Cho, S., Wang, J., Yang, M.H.: Deblurring low-light images with light streaks. In: CVPR (2014)
12. Jiang, H., Zheng, Y.: Learning to see moving objects in the dark. In: ICCV (2019)
13. Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., Yang, J., Zhou, P., Wang, Z.: Enlightengan: Deep light enhancement without paired supervision. TIP **30**, 2340–2349 (2021)
14. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: ICCV (2021)
15. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: CVPR (2018)
16. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In: CVPR (2019)
17. Li, C., Guo, C., Han, L., Jiang, J., Cheng, M.M., Gu, J., Loy, C.C.: Low-light image and video enhancement using deep learning: A survey. arXiv:2104.10729 (2021)
18. Li, C., Guo, C., Loy, C.C.: Learning to enhance low-light image via zero-reference deep curve estimation. TPAMI (2021)
19. Li, D., Xu, C., Zhang, K., Yu, X., Zhong, Y., Ren, W., Suominen, H., Li, H.: Arvo: Learning all-range volumetric correspondence for video deblurring. In: CVPR (2021)
20. Li, X., Chen, C., Zhou, S., Lin, X., Zuo, W., Zhang, L.: Blind face restoration via deep multi-scale component dictionaries. In: ECCV (2020)
21. Liu, J., Xu, D., Yang, W., Fan, M., Huang, H.: Benchmarking low-light image enhancement and beyond. IJCV **129**(4), 1153–1184 (2021)
22. Liu, R., Ma, L., Zhang, J., Fan, X., Luo, Z.: Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In: CVPR (2021)

23. Lore, K.G., Akintayo, A., Sarkar, S.: Llnet: A deep autoencoder approach to natural low-light image enhancement. Pattern Recognition **61**, 650–662 (2017)
24. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv:1608.03983 (2016)
25. Lv, F., Lu, F., Wu, J., Lim, C.: Mbllen: Low-light image/video enhancement using cnns. In: BMVC (2018)
26. Ma, C., Yang, C.Y., Yang, X., Yang, M.H.: Learning a no-reference quality metric for single-image super-resolution. CVIU **158**, 1–16 (2017)
27. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. TIP **21**(9), 3952–3966 (2012)
28. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal processing letters **20**(3), 209–212 (2012)
29. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Lee, K.M.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: CVPRW (2019)
30. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (2017)
31. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. In: ICCV (2017)
32. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: ECCV (2020)
33. Shen, Z., Wang, W., Lu, X., Shen, J., Ling, H., Xu, T., Shao, L.: Human-aware motion deblurring. In: ICCV (2019)
34. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: CVPR (2017)
35. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: CVPR (2015)
36. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (2018)
37. Wang, R., Xu, X., Fu, C.W., Lu, J., Yu, B., Jia, J.: Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment. In: CVPR (2021)
38. Wang, R., Zhang, Q., Fu, C.W., Shen, X., Zheng, W.S., Jia, J.: Underexposed photo enhancement using deep illumination estimation. In: CVPR (2019)
39. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: CVPR (2021)
40. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In: ICCVW (2021)
41. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: ECCVW (2018)
42. Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. In: BMVC (2018)
43. Yang, W., Wang, S., Fang, Y., Wang, Y., Liu, J.: From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In: CVPR (2020)
44. Yang, W., Wang, W., Huang, H., Wang, S., Liu, J.: Sparse gradient regularized deep retinex network for robust low-light image enhancement. TIP **30**, 2072–2086 (2021)
45. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Cycleisp: Real image restoration via improved data synthesis. In: CVPR (2020)

46. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Learning enriched features for real image restoration and enhancement. In: ECCV (2020)
47. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR (2021)
48. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR (2019)
49. Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R.W., Yang, M.H.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: CVPR (2018)
50. Zhang, K., Liang, J., Van Gool, L., Timofte, R.: Designing a practical degradation model for deep blind image super-resolution. In: ICCV (2021)
51. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: CVPR (2020)
52. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
53. Zhang, Y., Guo, X., Ma, J., Liu, W., Zhang, J.: Beyond brightening low-light images. IJCV **129**(4), 1013–1037 (2021)
54. Zhang, Y., Zhang, J., Guo, X.: Kindling the darkness: A practical low-light image enhancer. In: ACM MM (2019)
55. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
56. Zheng, C., Shi, D., Shi, W.: Adaptive unfolding total variation network for low-light image enhancement. In: CVPR (2021)
57. Zhou, S., Zhang, J., Pan, J., Xie, H., Zuo, W., Ren, J.: Spatio-temporal filter adaptive network for video deblurring. In: ICCV (2019)
58. Zhou, S., Zhang, J., Zuo, W., Xie, H., Pan, J., Ren, J.S.: DAVANet: Stereo deblurring with view aggregation. In: CVPR (2019)

LEDNet: Joint Low-light Enhancement and Deblurring in the Dark

Supplementary Material

In this supplementary material, we provide additional details and results to the paper. In Sec. A, we first present the architecture details of some modules in our proposed LEDNet. Sec. B presents further analysis and discussions on our proposed LEDNet network, consisting of more analysis and results on CurveNLU, and loss function. In Sec. C, we provide more analysis on our data synthesis pipeline and LOL-Blur dataset. Finally, more visual comparisons on both LOL-Blur and real-world images are provided in Sec. D. In addition, we present a video demo to show the effectiveness of proposed LEDNet for dealing with real-world dark blurry videos.

A   Architecture Details

As shown in Fig. 4 in the main manuscript, we adopt the existing Residual Downsample/Upsample [46] and Pyramid Pooling Module (PPM) [55] in our LEDNet. For reading convenience, Fig. 11(a) and (b) provide the detailed structures of Residual Downsample/Upsample and PPM, which are the same as their original configurations.


Residual Downsampling module   Residual Upsampling module   Pyramid Pooling Module
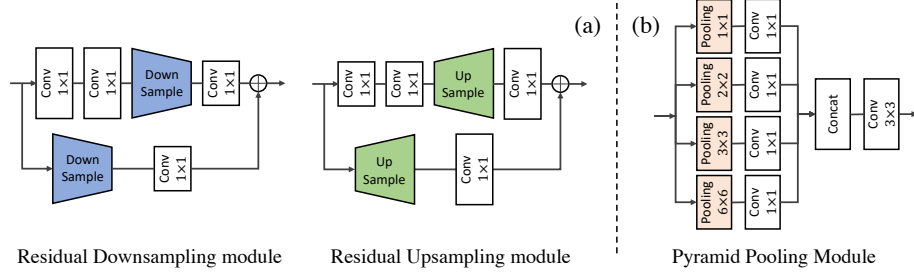
Fig. 11. Illustration of the Residual Downsample/Upsample Modules [46] and the Pyramid Pooling Module [55].

B   More Discussions on LEDNet

In this section, we first present more ablation experiments to show the effect of the key components of the proposed LEDNet, including CurveNLU, PPM, and enhancement loss.

B.1   Analysis on CurveNLU

Curve Parameter Visualization. To further explore CurveNLU, we visualize an example of estimated curve parameters A in Fig. 12(a). The parameters are significantly different between unsaturated regions and saturated regions. Fig. 12(b) shows two estimated curves of blue and red points in Fig. 12(a), which lay in the unsaturated region and saturated region respectively. The red curve of the unsaturated region has the greater curvature, thus, there is a lager feature intensity increase for darkder areas. In contrast, the blue curve of the saturated region that has a curvature close to 0 tends to maintain the feature value in the saturated regions. Therefore, the non-linear CurveNLU modules can increase intensity for dark areas while avoiding overexposure in the saturation regions.

(a) Curve Parameters A   (b) Estimated curves of two point positions
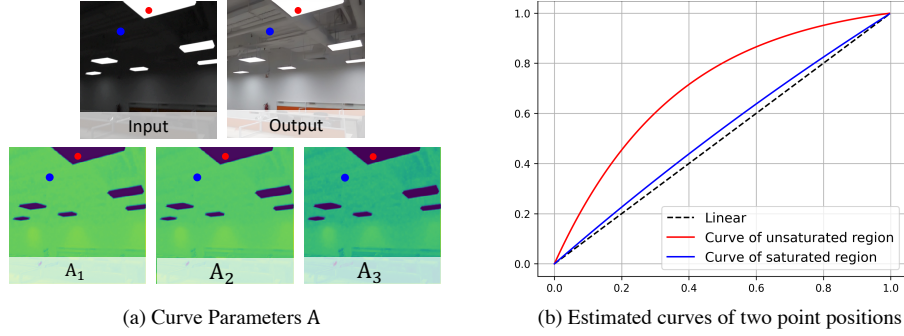
Fig. 12. (a) Visualization of estimated curve parameters. (b) Estimated curves of two points in the unsaturated region and saturated region.

Effect of Curve Order. To explore the effect of curve order n in the CurveNLU module, we conduct experiments that use different n for comparison. As shown in Table 4, using the higher curve order n over 3 only leads to slight PSNR/SSIM gains. Thus we use $n = 3$ in our CurveNLU modules as a trade-off between computational complexity and performance. Notably, compared with the baseline without CurveNLU inserted, i.e., $n = 0$, our proposed LEDNet obtains a large performance gain.

Table 4. Results on LOL-Blur dataset for different curve orders in CurveNLU modules.

	$n = 0$ (w/o CurveNLU)	$n = 1$	$n = 2$	$n = 3$ (Ours)	$n = 4$
PSNR	25.20	25.25	25.48	25.74	25.77
SSIM	0.823	0.826	0.838	0.850	0.850

B.2   Effectiveness of Enhancement Loss

Table 2 in the main manuscript has suggested that using enhancement loss $\mathcal{L}_{en}$ is indispensable in our method. Fig. 13 further shows removing the $\mathcal{L}_{en}$ in the training process harms the visual quality significantly. The network trained without $\mathcal{L}_{en}$ produces severe artifacts with unsmooth regions in the result.

<div align="center">Input   w/o Enhancement Loss   w/ Enhancement Loss</div>

Fig. 13. Visual comparison of different losses that using enhancement loss or not.

C   More Discussions on LOL-Blur Dataset
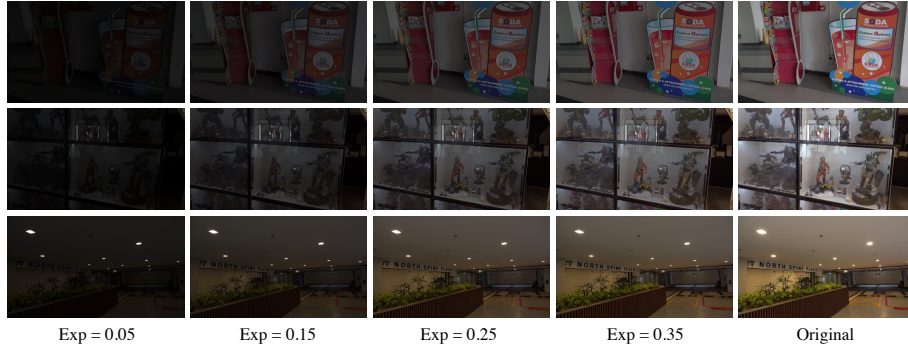
C.1   Simulation of Low light

In this paper, we use the Exposure Conditioned Zero-DCE (EC-Zero-DCE) to generate the low-light images of different exposure levels. Fig. 14(a) compares our low-light data synthesis pipeline with Gamma correction that has been used in prior works [23,25]. As we can see from this comparison, the image generated by Gamma correction has a large color deviation with noticeable warm tones. In contrast, our EC-Zero-DCE can produce more natural and realistic low-light images. Moreover, the proposed EC-Zero-DCE performs pixel-wise and spatially-varying light adjustment, Fig. 14(b) provides a non-uniform luminance adjustment map for this case.

Given different exposure levels, EC-Zero-DCE can generate realistic low-light images with diverse darkness, as shown in Fig. 15. Thanks to its spatially-varying adjustment, EC-Zero-DCE tends to retain the intensities of saturated pixels, simulating the realistic light effect in real-world night images.

| Normal-light image | Gamma Correction | EC-Zero-DCE (Ours) | (a) | (b) Non-uniform Adjustment Map (EC-Zero-DCE) |

Fig. 14. (a) Comparisons between the proposed EC-Zero-DCE and Gamma correction for generating low-light images. (b) The non-uniform adjustment map suggests that CE-Zero-DCE performs spatially-varying luminance adjustment.

| Exp = 0.05 | Exp = 0.15 | Exp = 0.25 | Exp = 0.35 | Original |

Fig. 15. The proposed EC-Zero-DCE is able to generate low-light images with different exposure levels, while keeping the intensities of saturated pixels unchanged.

C.2   Simulation of Noise

To simulate realistic noise in dark images, we adopt CycleISP [45] to generate the noisy image in the RAW domain. We compare our noise simulation with Gaussian and Poisson noise that are commonly used in other restoration tasks, e.g., blind face restoration [20,39] and real-world blind super-resolution [40,50,1]. Fig. 16 shows the noises generated by CycleISP are more natural and realistic.

| Low-light clean image | Gaussian noise | Poisson noise | Realistic noise using CycleISP |
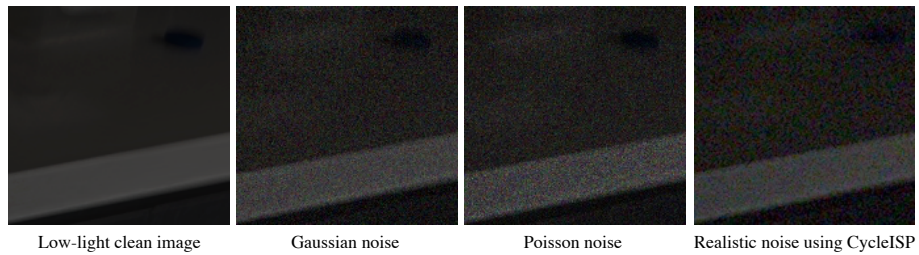
Fig. 16. Noise simulation comparison among CycleISP, Gaussian, and Poisson.

C.3   Luminance Distribution of Datasets

Fig. 17(a) shows the luminance distribution of our proposed LOL-Blur Dataset. Fig. 17(b) provides a comparison of luminance distributions of different deblurring datasets. The great majority brightness of ground truth images in the RealBlur dataset lay the range of small intensity, thus, RealBlur is not suitable for training a light enhancement network. Besides, there are many sunny scenes in the REDS dataset, which can not be adopted to generate low-light images.

(a) Luminance distribution of LOL-Blur dataset
(Input and GT)

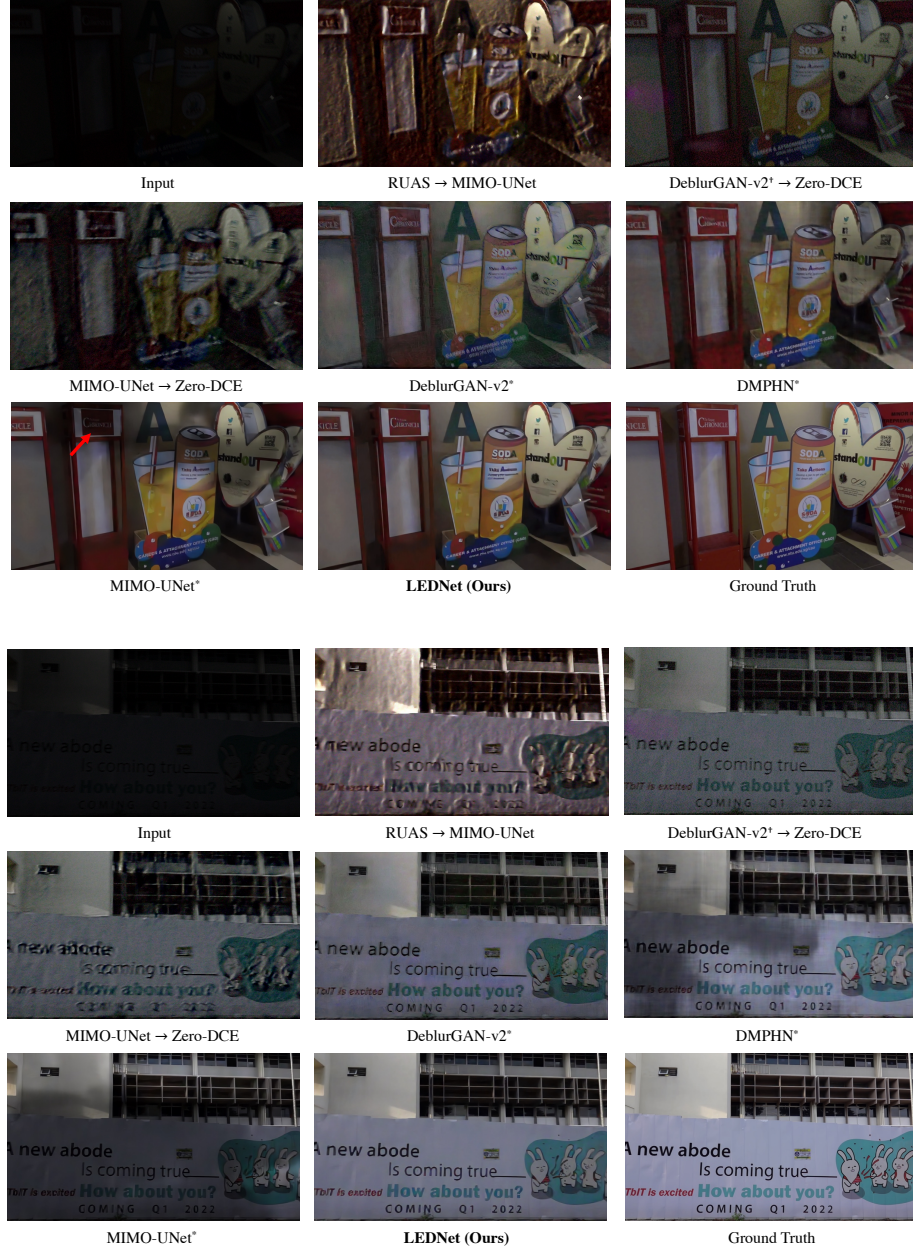(b) Luminance distribution of different datasets
(GT)
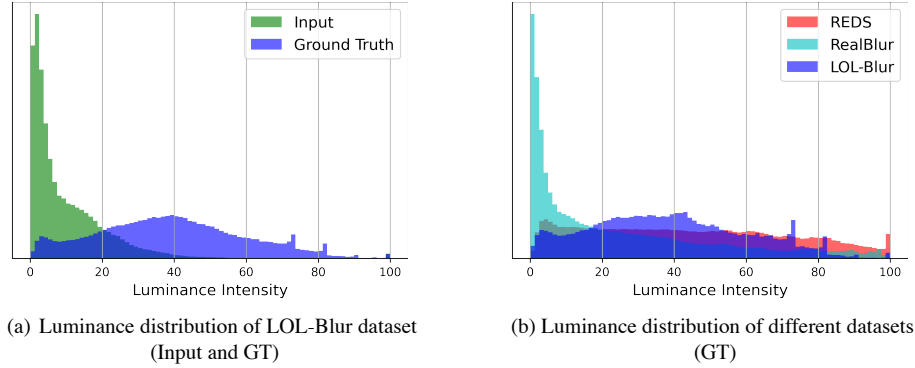
Fig. 17. (a) Luminance distribution of LOL-Blur dataset. (b) Comparison of Luminance distributions of different deblurring datasets.
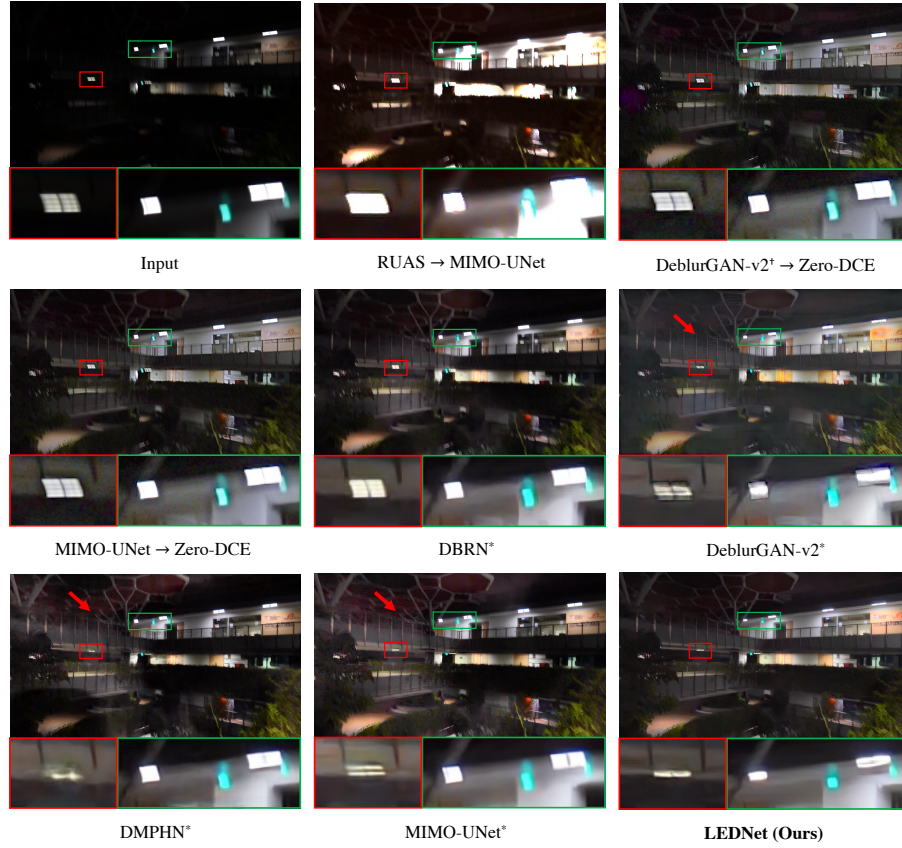
D   Qualitative Comparisons

In this section, we present more visual comparisons with the baselines studied in the main manuscript: RUAS [22] $\rightarrow$ MIMO-UNet [7], DeblurGAN-v2† [16] $\rightarrow$ Zero-DCE [9], MIMO-UNet [7] $\rightarrow$ Zero-DCE [9], DRBN* [43], DeblurGAN-v2* [16], DMPHN* [48], and MIMO-UNet* [7]. Fig. 18 provides more visual comparisons on our LOL-Blur Dataset. In addition, Figs. 19, 20, and 21 provide more visual comparisons on real-world night blurry images. To demonstrate the generalizability in the wild of our dataset and network, we also test on more real-world night blurry images in the RealBlur dataset [32]. Fig. 22 shows our LEDNet is able to handle various blur patterns (revealed by light streaks in the input images). Besides, Fig. 23 provides more results in different scenarios on the RealBlur dataset. The above extensive real-world results[1] suggest that our method performs well on diverse test images and videos in the wild.

---

[1] Note that all the Figs. 19, 20, 21, 22, and Fig. 23 provided in this suppl., as well as all results in [video demo] are evaluated on the real-world scenarios.

Input	RUAS → MIMO-UNet	DeblurGAN-v2† → Zero-DCE
MIMO-UNet → Zero-DCE	DeblurGAN-v2*	DMPHN*
MIMO-UNet*	**LEDNet (Ours)**	Ground Truth

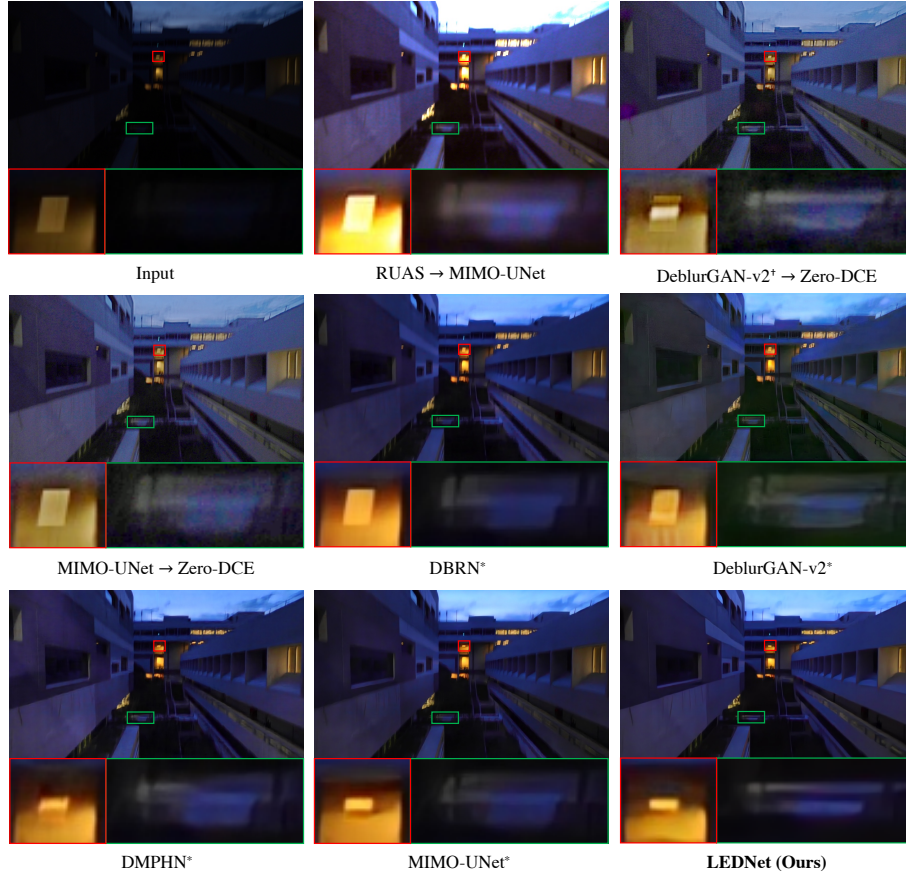|		
Input	RUAS → MIMO-UNet	DeblurGAN-v2† → Zero-DCE
MIMO-UNet → Zero-DCE	DeblurGAN-v2*	DMPHN*
MIMO-UNet*	**LEDNet (Ours)**	Ground Truth

Fig. 18. Visual comparison on our LOL-Blur dataset. The proposed LEDNet generates much sharper images with visually pleasing results. The symbol '†' indicates that we use DeblurGAN-v2 trained on RealBlur dataset, and '*' indicates the network is trained with our LOL-Blur dataset. **(Zoom in for best view)**

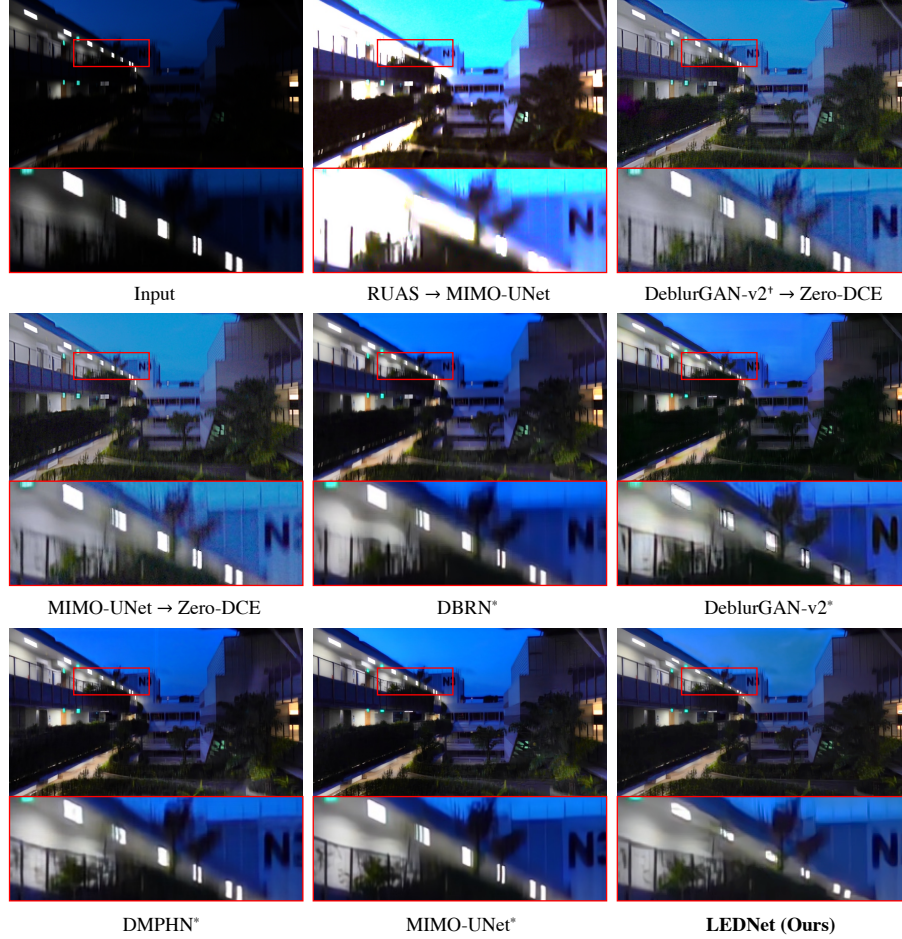|		
Input	RUAS → MIMO-UNet	DeblurGAN-v2† → Zero-DCE
MIMO-UNet → Zero-DCE	DBRN*	DeblurGAN-v2*
DMPHN*	MIMO-UNet*	**LEDNet (Ours)**

Fig. 19. Visual comparison on a real-world night blurry image. The proposed LEDNet achieves the best perceptual quality with more stable light enhancement and better deblurring performance, especially in saturated regions, while other methods still leave large blurs in saturated regions and suffer from noticeable artifacts, as indicated by red arrows. The symbol '†' indicates that we use DeblurGAN-v2 trained on RealBlur dataset, and '*' indicates the network is trained with our LOL-Blur dataset. (**Zoom in for best view**)

Input	RUAS → MIMO-UNet	DeblurGAN-v2[†] → Zero-DCE
MIMO-UNet → Zero-DCE	DBRN*	DeblurGAN-v2*
DMPHN*	MIMO-UNet*	**LEDNet (Ours)**

Fig. 20. Visual comparison on a real-world night blurry image. The proposed LEDNet achieves the best perceptual quality with more stable light enhancement and better deblurring performance, especially in saturated regions, while other methods still leave large blurs in saturated regions and suffer from noticeable artifacts. The symbol '[†]' indicates that we use DeblurGAN-v2 trained on RealBlur dataset, and '*' indicates the network is trained with our LOL-Blur dataset. **(Zoom in for best view)**

| Input | RUAS → MIMO-UNet | DeblurGAN-v2† → Zero-DCE |

| MIMO-UNet → Zero-DCE | DBRN* | DeblurGAN-v2* |

| DMPHN* | MIMO-UNet* | **LEDNet (Ours)** |

Fig. 21. Visual comparison on a real-world night blurry image. The proposed LEDNet achieves the best perceptual quality with more stable light enhancement and better deblurring performance, especially in saturated regions, while other methods still leave large blurs in saturated regions and suffer from noticeable artifacts. The symbol '†' indicates that we use DeblurGAN-v2 trained on RealBlur dataset, and '*' indicates the network is trained with our LOL-Blur dataset. **(Zoom in for best view)**

Fig. 22. Evaluation on different blur patterns on RealBlur dataset [32]. The proposed LEDNet is able to handle blurs of different shapes, which can be observed from the light streaks in the input images. **(Zoom in for best view)**

Real night blurry image   LEDNet   Real night blurry image   LEDNet
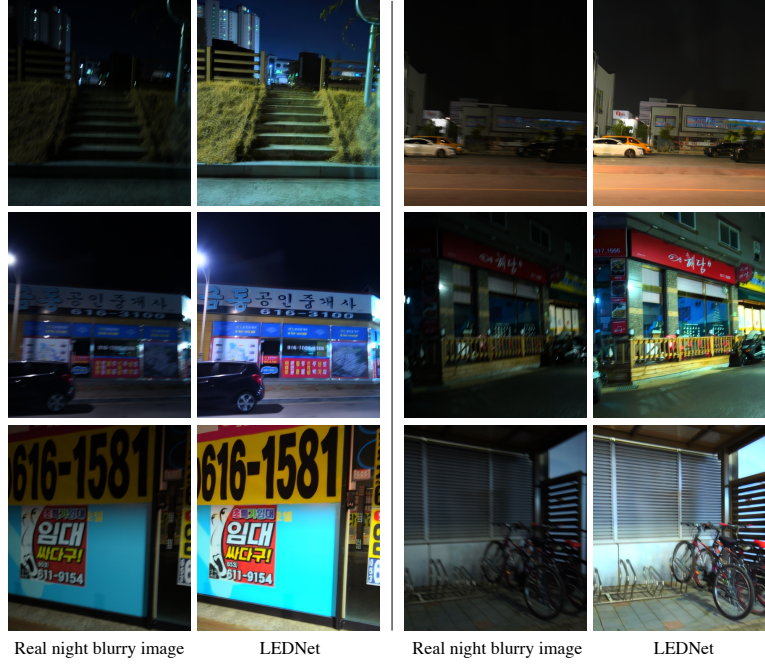
Fig. 23. Visual results on RealBlur dataset [32]. The proposed LEDNet performs well in the different scenarios. **(Zoom in for best view)**