D2HNet: Joint Denoising and Deblurring with Hierarchical Network for Robust Night Image Restoration

Yuzhi Zhao[1], Yongzhe Xu[2], Qiong Yan[2], Dingdong Yang[2], Xuehui Wang[3], and
Lai-Man Po[1]

[1] Department of Electrical Engineering, City University of Hong Kong
[2] SenseTime Research and Tetras.AI
[3] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
`yzzhao2-c@my.cityu.edu.hk, wangxuehui@sjtu.edu.cn, eelmpo@cityu.edu.hk,`
`{xuyongzhe1,yanqiong,yangdingdong}@tetras.ai`

Abstract. Night imaging with modern smartphone cameras is troublesome due to low photon count and unavoidable noise in the imaging system. Directly adjusting exposure time and ISO ratings cannot obtain sharp and noise-free images at the same time in low-light conditions. Though many methods have been proposed to enhance noisy or blurry night images, their performances on real-world night photos are still unsatisfactory due to two main reasons: 1) Limited information in a single image and 2) Domain gap between synthetic training images and real-world photos (e.g., differences in blur area and resolution). To exploit the information from successive long- and short-exposure images, we propose a learning-based pipeline to fuse them. A D2HNet framework is developed to recover a high-quality image by deblurring and enhancing a long-exposure image under the guidance of a short-exposure image. To shrink the domain gap, we leverage a two-phase DeblurNet-EnhanceNet architecture, which performs accurate blur removal on a fixed low resolution so that it is able to handle large ranges of blur in different resolution inputs. In addition, we synthesize a D2-Dataset from HD videos and experiment on it. The results on the validation set and real photos demonstrate our methods achieve better visual quality and state-of-the-art quantitative scores. The D2HNet codes and D2-Dataset can be found at `https://github.com/zhaoyuzhi/D2HNet`.

Keywords: Night Image Restoration, Image Denoising, Image Deblurring, Domain Gap Issue

1   Introduction

Capturing high-quality photos at night-time on modern smartphones is troublesome due to the limitations of sensors and optical systems. It is a long-standing and practical problem in the computational photography field. Acquiring sharp and clean photos effectively and efficiently on smartphones in night conditions is

in great demand. The main difficulty lies in that the image signal is too weak compared with the inherent noise in the imaging process, which yields a low signal-to-noise ratio (SNR) and degrades image quality [41, 42, 70]. To obtain higher SNR, there are many solutions either on the hardware level (in-camera solutions) or algorithm level, which typically fall into one of these three categories: 1) *Physical solutions*: using a larger sensor, opening the aperture, using flash, or setting longer exposure time; 2) *Single-image restoration*: deblurring the long-exposure image with motion blurs, or denoising the short-exposure image with severe noises; 3) *Burst-image restoration*: combining several photos captured in quick succession using temporal coherence within the burst.

Though these solutions improve the night image restoration quality, they might not meet the requirements of both *effectiveness* and *efficiency* for mobile photography. For *physical solutions*, larger sensor size and aperture are related to hardware design and increase the cost. The built-in flash does not help for far scenes. Long exposure time causes motion blur. To post-process the captured images, *single-image restoration* methods have been widely studied, e.g., training neural networks [36, 90] on a large number of paired degraded-clean images. However, a single input image contains limited information thus restricting the restoration quality. To use more information, *burst-image restoration* methods [46, 49] combine multiple continuous frames to generate a single good image. Though they have a theoretically superior SNR than single-image restoration methods, the speed is restricted by the capturing process, including multiple exposure and readout time. Meanwhile, the misalignment issue has to be solved for all captured frames. In addition, the data distribution gap (e.g., differences in blur area, resolution) between training and real images remains a key problem.

In this paper, we tackle this real-world problem by post-processing successive long- and short-exposure images through a D2HNet framework. It can produce clean and sharp photographs on mobile devices without any manual control or extra hardware support. Compared with the previous image restoration approaches, our approach has three main advantages: 1) Taking advantage of both long and short exposures; 2) Addressing the domain gap issue between training data and real-world photos by a special two-phase network; 3) Balancing image processing quality and capturing time (only 2 long- and short-exposure images are needed).

Normally, long-exposure images have regular color and fewer noises and short-exposure images are of trivial blurs. Compared with single-image restoration [36, 90], fusing them helps reduce the noise level and blurriness, and improve color fidelity for night photos. Compared with burst-image restoration methods [46, 49], two shots have milder misalignment issues and require a shorter capturing time. Built upon these observations, we propose the D2HNet framework. To address the domain gap between training data and real photos captured by different smartphones, we split D2HNet into sequential subnets: *DeblurNet* and *EnhanceNet*. DeblurNet runs on a small resolution at the training and a fixed resolution at the testing similar to training images. It ensures pixel shifts or blur levels between training and real-world images are comparable; therefore, the network generalizes better to different image resolutions and blur areas.
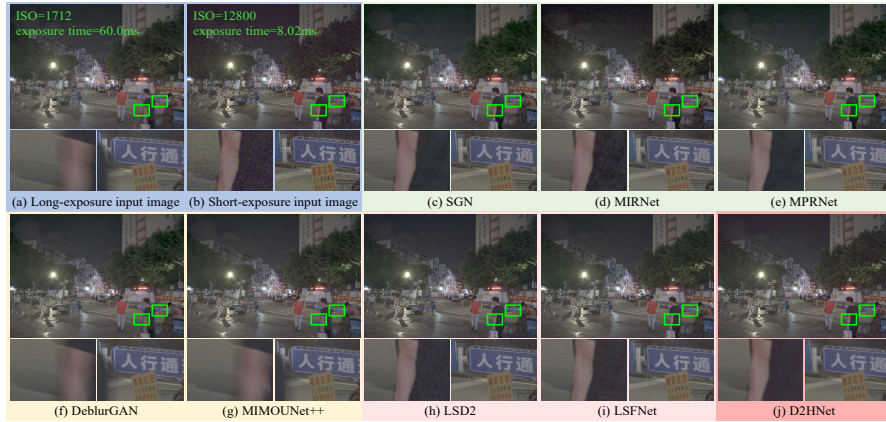
ISO=1712 exposure time=60.0ms	ISO=12800 exposure time=8.02ms			
(a) Long-exposure input image	(b) Short-exposure input image	(c) SGN	(d) MIRNet	(e) MPRNet
(f) DeblurGAN	(g) MIMOUNet++	(h) LSD2	(i) LSFNet	(j) D2HNet

Fig. 1. Performance on real-world night photos. Input photos captured by *Xiaomi Mi Note 10* smartphone are shown in (a) and (b). Results from single image denoising methods are in (c) - (e). Results from single image deblurring methods are in (f) and (g). Results from long-short fusion methods (including our D2HNet) are in (h) - (j).

EnhanceNet enhances the DeblurNet output on actual target resolutions together with short- and long-exposure inputs. We use deformable convolutions [18] in the EnhanceNet to align the features hierarchically to better extract the details and textures from the long-exposure input. In addition, we propose a *CutNoise scheme* to assist the learning of where and how to deblur and a *VarmapSelection scheme* to balance blurry and non-blurry patches during training.

To evaluate the capability of D2HNet, we synthesize a D2-Dataset from HD videos for training and validation. It contains 6853 tuples of long- and short-exposure images with corresponding sharp ground truth. It covers a wide range of scenes, e.g., cities, villages, forests, deserts, and mountains. We also capture 28 pairs of long- and short-exposure photos of real-world scenes by a smartphone for testing. Extensive experiments on both D2-Dataset and real captured photos show the state-of-the-art (SOTA) performance achieved by the proposed D2HNet. One real sample is shown in Figure 1. D2HNet can produce clean and sharp images simultaneously, while the other methods fail to do so. It demonstrates that D2HNet better utilizes the information of dual-exposure images.

Below we summarize the main contributions of this paper:

1) We propose a two-phase D2HNet for robust real night image restoration and to address the domain gap issue between training data and real photos;

2) We propose two data augmentation schemes, CutNoise and VarmapSelection, to improve and stabilize the training of D2HNet;

3) We create a D2-Dataset including 6853 image tuples with multiple levels of blurs for benchmarking D2HNet;

4) We conduct extensive experiments with long-short fusion methods, and single image denoising or deblurring methods. The proposed D2HNet achieves better performance than other methods.

2   Related Work

Single-image Denoising. Image denoising is a fundamental topic in image processing. Previous methods such as total variation [59], wavelet coring [60], non-local means [4], BM3D [17] assumed noises and signals have specific statistical regularities. However, these methods used hand-crafted models thus not robust to real noises. Recently, CNNs have shown their advanced performance to address blind denoising issue [13, 25, 43, 44, 48, 65, 90, 91]. Some works further extended them to reduce real noises [2, 5, 10–12, 14, 26, 29, 33, 38, 45, 82, 84, 85, 89, 93]. To better simulate noise emerged on mobile ISP, many inverse algorithms [3, 77, 83] and real noise calibration methods [1, 71, 73] were proposed.

Single-image Deblurring. Image deblurring aims to generate a sharp and clean reconstruction from a blurry input. Many classical non-blind methods formulated the problem as blind deconvolutions [35, 39, 56]. The blur kernels are normally assumed as noisy linear operators enforced on the clean images. Recently, CNN-based approaches [6, 15, 21, 23, 30, 36, 37, 51–54, 57, 63, 64, 66, 74, 81, 88, 92] proposed the end-to-end deblurring with specific network architectures and loss functions. These methods are trained on large-scale blurry-sharp pairs. However, directly applying them to real-world photos may not obtain sharp results.

Burst-image Restoration. Since the overall photon counts of burst images are more than a single image, burst-image-based methods [20, 22, 32, 46, 49, 76, 78, 86] have theoretically superior SNR than single-image-based methods. However, burst images suffered from noises and camera shake, which increase the difficulty of implementation. To overcome that, [22] proposed a recurrent neural network to filter noises in a sequence of images. [49] combined neural network and kernel method to perform denoising and alignment jointly. Though they restore high-quality photos, their data capture occupies a major time during application.

Image Restoration by Fusing Successive Long- and Short-exposure images. Image restoration with dual exposures [16,24,61,62,68,75,80] is beneficial for both noise reduction and blur estimation. For instance, Yuan et al. [80] firstly estimated blur kernels using the texture of short-exposure images, which are then used to restore the long-exposure blurry images. Recently, LSD2 [50] and LSFNet [9] used CNNs to fuse dual-exposure images and obtained better results than single-image denoising or deblurring methods on their synthetic dataset. However, they ignored the potential domain gap issue between training images and real-world photos.

Deformable Convolution. Dai et al. [18] proposed deformable convolutions, which allows the network to obtain the information away from regular local neighborhoods by learning additional offsets. It has been widely applied in computer vision tasks such as semantic segmentation [18,94], video deblurring [69], video super-resolution [7,8,67], and video restoration [19,27]. For instance, EDVR [69] used deformable convolutions to align inputs without using explicit optical flows. For the long-short fusion problem, there normally exists a misalignment issue between input long- and short-exposure images. Also, it is difficult to compute accurate optical flows from noisy and blurry inputs. Inspired by previous methods, we adopt deformable convolutions as alignment blocks.

3   Data Acquisition

D2-Dataset. We synthesize a D2-Dataset for training and benchmarking. The data synthesis pipeline is as follows:

1) We collect 30 HD videos with 1440×2560 resolution from the Internet. They are almost noise-free and cover a wide range of scenes. We sample 60 continuous frames (approximately 1 second in original 60-fps videos) every 10 seconds in each video to reduce repeated scenes and avoid scene switching;

2) We use a video frame interpolation model SuperSloMo [31] to increase the original 60-fps videos to 960 fps. It smooths videos to simulate realistic blurs;

3) We synthesize successive long-exposure image l and short-exposure image s by averaging interpolated frames. Meantime, we add a time gap between l and s to model hardware readout limitation. We also extract corresponding sharp single frames, i.e., the last frame of long-exposure image l_{last} and the first frame of short-exposure image s_{first}.

The pipeline results in 6853 image tuples (l, s, l_{last}, and s_{first}), where 5661 tuples are used for training and 1192 for validation. More details are presented in the supplementary material.

Testing Images. We capture 28 pairs of long- and short-exposure images with resolution 3472×4624 using a *Xiaomi Mi Note 10 smartphone*. To ensure the overall brightness of long- and short-exposure images are approximately the same, we set "ISO×exposure time" of them equal. Specifically, the exposure time of the long-exposure image is set to be 8 times of short-exposure time while its ISO is 1/8 of the short-exposure image.

4   Methodology

4.1   Problem Formulation

Given paired noisy long- and short-exposure images denoted as l_n and s_n, we aim to recover a sharp and clean image z. We formulate it as maximizing a posteriori of the output conditioned on inputs and D2HNet parameters Θ:

$$\Theta^* = \arg\max_{\Theta} p(z|l_n, s_n, \Theta). \tag{1}$$

We train our network on the proposed D2-Dataset. From it we use l, s pair to generate noisy training inputs l_n, s_n, and s_{first} as ground truth of z here.

4.2   D2HNet Architecture and Optimization

Workflow. The workflow of D2HNet is illustrated in Figure 2. To address the domain gap issue between synthetic training images and real-world photos (e.g., different resolutions and blur levels), we use a two-phase structure in our workflow. In phase one, the two inputs are downsampled into a smaller and fixed resolution, so that motion scales and pixel shifts in the long-exposure image
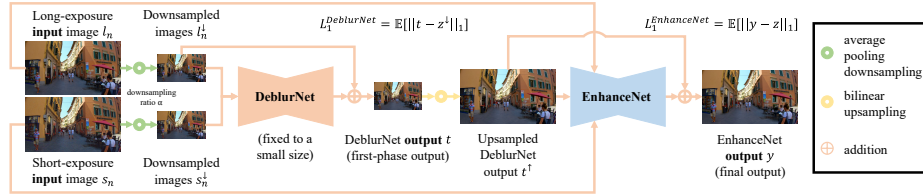
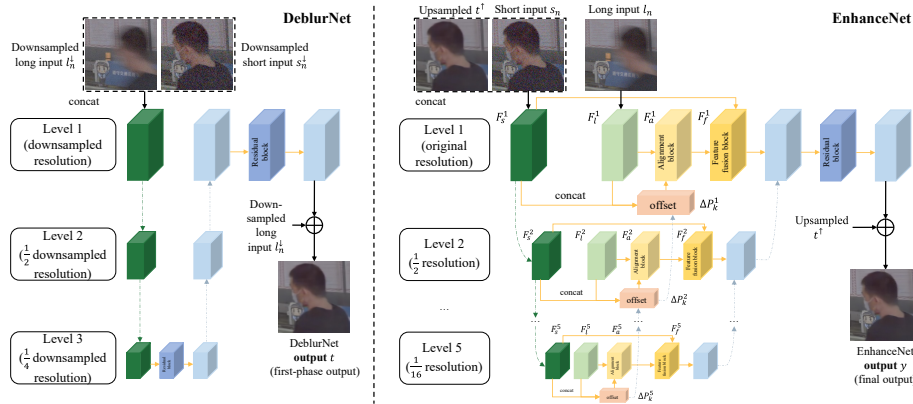**Fig. 2.** Illustration of the D2HNet workflow.

Fig. 3. Illustration of DeblurNet (left) and EnhanceNet (right). Alignment blocks, Feature fusion blocks, and Residual blocks are noted in the figure while the remaining blocks are normal convolutional layers. Offsets are learnable and as parts of Alignment blocks. Images t and y are the outputs of DeblurNet and EnhanceNet, respectively.

are restricted. Then, the DeblurNet with a certain receptive field can perform accurate deblurring based on the edge information in the short-exposure image. In phase two, to enhance the details lost during downsampling, the EnhanceNet post-process the upsampled first-phase network output together with long-short inputs in their original resolution.

DeblurNet. For the input data, DeblurNet receives the downsampled long- and short-exposure input images with a downsampling ratio α. At the training, α is set to $1/2$; while at the testing, the input resolution is fixed to 1024×1024. Therefore, the domain gap is reduced since the motion scales of testing images are controlled. We use the average pooling as the downsampling operator at both training and testing stages as it mimics the physical differences between low- and high-resolution images better. For the network architecture, DeblurNet uses 3 levels to extract features since it better balances the deblurring quality and computational complexity, where DWT [43] is used as the downsampling operator. There are two Residual blocks at the bottleneck and at the tail respectively, where each block includes 4 sequential residual layers [28]. The output t is upsampled by bilinear sampling and then as the input for the next phase, i.e., EnhanceNet.

EnhanceNet. The target of EnhanceNet is to recover the details (mostly from long-exposure input) and further remove artifacts for the upsampled DeblurNet output $t^{\uparrow}$. As shown in Figure 3, EnhanceNet has 3 modules: feature pyramid extraction (green blocks), alignment and feature fusion (pink and yellow blocks), and reconstruction (blue blocks).

The feature pyramid extraction has two branches without sharing weights. The output two feature pyramids have 5 levels, denoted as F_s^1-F_s^5 and F_l^1-F_l^5, respectively. Since inputs $s_n/y^{\uparrow}$ and l_n are not spatially aligned, we perform the alignment for long-exposure features (F_l^1-F_l^5) by Alignment blocks, where we use the modulated deformable convolution [94]. Alignment blocks allow the following layers to better fuse the information of two feature pyramids. Here we give a brief introduction for the modulated deformable convolution. As we have known, a 3×3 convolution kernel of dilation 1 has learnable weights $w_k \in \{1, ..., K\}$ and fixed offsets $p_k \in \{(-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,1),(1,-1),(1,0),(1,1)\}$, where $K = 9$. Then for the modulated deformable convolution, there are learnable parameters, offsets Δp_k^i and modulation scalars Δm_k^i for each location p_k. The offsets Δp_k^i are real numbers and the modulation scalars Δm_k^i are in range of [0, 1]. Therefore, for such convolution result on i-th long-exposure feature F_l^i can be expressed as:

$$F_a^i(p) = \sum_{k=1}^{K} w_k^i \cdot F_l^i(p + p_k + \Delta p_k^i) \cdot \Delta m_k^i. \tag{2}$$

The modulation scalars and learnable offsets (pink blocks in Figure 3) are learned from short- and long-exposure features hierarchically. For simplicity, we only express the learnable offsets from a series of convolutional layers c^i as:

$$\Delta P_k^i = \begin{cases} c^i(F_s^i, F_l^i, \Delta P_k^{i+1}), & i = 1, 2, 3, 4 \\ c^i(F_s^i, F_l^i), & i = 5 \end{cases} \tag{3}$$

where $\Delta P_k^i = \{\Delta p_k^i\}$. The deepest 5-th level ΔP_k^5 is first computed. Since the misalignment between two features F_s^i and F_l^i of the deepest level is small [21], the learning of offsets is relatively accurate and less challenging. We then pass the learned offsets to the upper levels to learn more precise offsets. This process is done level-by-level as a hierarchical refinement [9, 47, 69].

After long-exposure features F_l^1-F_l^5 are aligned, we perform the feature fusion. In i-th level's Feature fusion block, the aligned features F_a^i and short-exposure features F_s^i are concatenated and then processed by a Residual block r^i as:

$$F_f^i = r^i(F_s^i, F_a^i), \tag{4}$$

where the output features F_f^5 are connected to the first decoder layer, while F_f^1-F_f^4 serve as short-cut connections like in UNet [58]. Finally, we use a Residual block (including 4 residual layers) at the tail to further refine the features at the original resolution. The final output is added to $t^{\uparrow}$.

Loss. We first train DeblurNet and then train EnhanceNet. The L1 loss [90] is used for training them, as shown in Figure 2. They are expressed as:

$$L_1^{DeblurNet} = \mathbb{E}[||t - z^{\downarrow}||_1], \ L_1^{EnhanceNet} = \mathbb{E}[||y - z||_1], \tag{5}$$
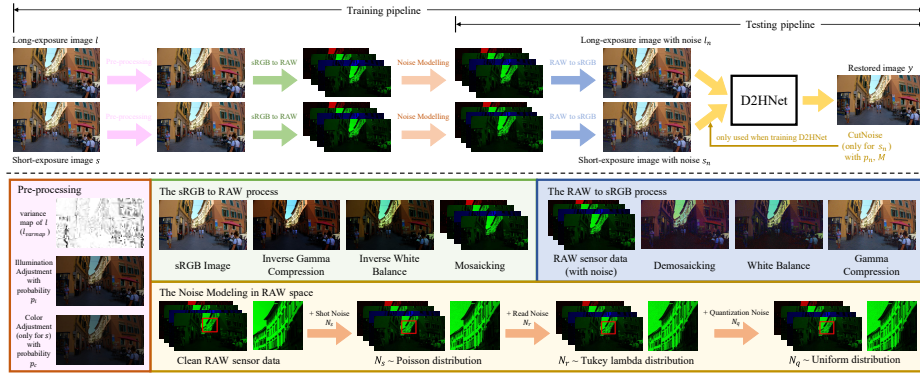
Fig. 4. Illustration of detailed procedures of training and testing pipelines (upper) and examples of every step of data processing procedures (lower).
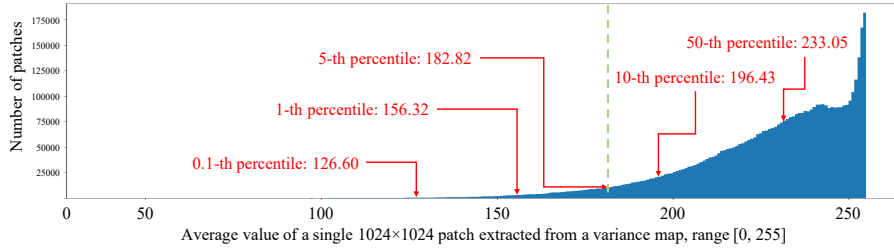

Fig. 5. Variance map patch distribution on D2-Dataset. The 5-th percentile point 182.82 is set as the threshold. If the average of a variance map patch is lower than 182.82, its corresponding long-exposure patch is regarded to be highly blurry.

where $z^{\downarrow}$ is the average pooling downsampled result from the ground truth z to match the resolution of t.

4.3   Data Processing

The data processing is a key process for training the D2HNet, which includes VarmapSelection, Appearance Adjustment, Noise Modeling, and CutNoise, as shown in Figure 4. It is designed to better model the real image distortion and balance the training data distribution.

VarmapSelection. It is a variance-map-based selection scheme to address the imbalanced blurriness issue and improve the robustness of D2HNet. Since most regions of the simulated long-exposure image l are of low levels of blur, simply randomly choosing patches at the training causes loss fluctuation and ineffective deblurring ability for large motion. VarmapSelection scheme uses the variance map to represent the blur level and choose larger motion patches for training. The variance map is calculated from both l and l_{last} as:

$$l_{varmap} = \min(\mathrm{Var}(l))/\mathrm{Var}(l_{last}), 1), \tag{6}$$

where Var computes the variance using a $k \times k$ window, with stride k. The resulted map is $1/k \times 1/k$ of the original size, so we upsample it with the nearest interpolation. According to the definition, a smaller value means higher levels of blur. One example variance map is shown in the pink rectangle of Figure 4, where the blurrier regions in l have clearer (darker) responses in l_{varmap}.

To determine whether a randomly chosen patch is of a large blur, we choose to define a threshold based on variance statistics on the training set. For each variance map, we randomly sample 1000 different squares of size 1024×1024 and calculate the average variance value for each square. Then, we sort all values across the dataset and use the 5-th percentile point as the threshold, as shown in Figure 5. Afterward, we do sampling again, keep only the squares that have lower average variance values than the threshold and draw patches from the corresponding long-short-GT tuples. This process results in additional 9453 tuples of a strong blur. They are added to the original training set.

Appearance Adjustment. To simulate low-light image tuples, we apply Illumination Adjustment (IA) to lower the overall brightness. We also apply Color Adjustment (CA) to model the difference between long- and short-exposure images. An example is shown in the pink rectangle of Figure 4. IA is done by an inverse gamma compression for long-short-GT tuples, as follows:

$$IA(u) = \max(u, \varepsilon)^g, \text{ for } u \in \{s, l, z\}, \tag{7}$$

where $\varepsilon = 10^{-8}$. The gamma value g is randomly chosen from $[1/0.6, 1/0.7, 1/0.75, 1/0.8, 1/0.9]$. CA is achieved by a linear transform to disturb the overall color and brightness for only the short-exposure image s. It is defined as:

$$CA(s) = a \cdot s + b, \tag{8}$$

where a and b are sampled uniformly from $[0.3, 0.6]$ and $[0.001, 0.01]$, respectively.

Noise Modeling. We calibrate real smartphone noises in the RAW image space following [73] and then apply the noise simulation. Since our D2-dataset contains only sRGB images, we adopt a simple reverse ISP process [3] to convert them from sRGB to RAW. It includes an inverse gamma compression (as in Equation 7 with $g = 2.2$), an inverse white balance which simply scales R and B channels by scalar $1/w_r, 1/w_b$ separately, and the mosaic to form Bayer pattern. Note white balance gain for the G channel (w_g) is fixed to 1, while w_r and w_b are sampled uniformly from $[1.9, 2.4]$ and $[1.5, 1.9]$, respectively. After noise modeling, we then convert it to sRGB by a forward ISP process with corresponding parameters for inversion. The noise model is calibrated on the *Xiaomi Mi Note 10*, which we use to capture real photos.

CutNoise. To encourage the fusion and utilization of the short-exposure image, inspired by [79], we design the CutNoise scheme. It is performed after noise simulation on the short-exposure image which has stronger noise. CutNoise randomly selects a region and copies ground truth z (i.e., s_{first}) to the corresponding position of s_n. The region itself can be any shape but we fix it to square for easy implementation. With CutNoise, D2HNet will not degenerate to use only the blurrier long-exposure input but is forced to learn to fuse information from the sharper short-exposure one, therefore generating sharper output.

Table 1. Comparisons of D2HNet and other methods on D2-Dataset validation set by PSNR and SSIM [72]. The red and blue colors denote the best and second-best results, respectively.

Table 2. The results of the human perceptual study on real photos for the D2HNet and other pipelines, given by preference rates (PR) for D2HNet over all the votes.

Method	1440p val data		2880p val data	
	PSNR	SSIM	PSNR	SSIM
DenseFuse [40]	32.90	0.9484	34.70	0.9637
LSD2 [50]	33.20	0.9517	35.36	0.9675
LSFNet [9]	33.87	0.9557	36.17	0.9715
DeblurGAN [36]	33.80	0.9558	36.26	0.9701
SGN [25]	33.87	0.9567	36.25	0.9720
TP1	34.35	0.9628	36.66	0.9755
TP2	34.26	0.9599	36.54	0.9733
TP3	34.41	0.9611	36.70	0.9747
D2HNet	34.67	0.9639	36.85	0.9767

Method	PR
D2HNet > DenseFuse, LSD2, LSFNet DeblurGAN, SGN	86.07%
D2HNet > TP1	82.50%
D2HNet > TP2	81.07%
D2HNet > TP3	79.28%

5   Experiment

5.1   Implementation Details

Our training samples include original 5661 tuples of full-resolution images from the D2-Dataset and 9453 tuples of strong blurry patches selected by the VarmapSelection scheme. For DeblurNet, the input resolution is fixed to $512{\times}512$ by average pooling. The epochs are 100 and the learning rate is initialized as 1×10^{-4}. For EnhanceNet, the input resolution is $256{\times}256$ randomly cropped patches due to memory limit. The epochs are 150 and the learning rate is initialized as 5×10^{-5}. For both subnets, the learning rates are halved every 50 epochs. The batch size equals 2 and an epoch includes 5661 iterations, corresponding to the number of training tuples. The Adam optimizer [34] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ is used. The probabilities of performing Illumination Adjustment, Color Adjustment, and CutNoise are set to 0.3, 0.5, and 0.3, respectively. The size of the CutNoise square is 120. We implement the D2HNet with PyTorch 1.1.0 and train it on 2 Titan Xp GPUs. It takes approximately 2 weeks to complete the optimization.

5.2   Long-short Fusion Method Experiments

We compare the image restoration quality of D2HNet and other recent works with similar target, DenseFuse [40], LSD2 [50], and LSFNet [9], or with SOTA performance in either denoising or deblurring, SGN [25], DeblurGAN [36] (see more in Section 5.3). To fit the dual inputs, SGN's and DeblurGAN's input layers are changed to receive two images. In addition, we define three more two-phase pipelines for a more comprehensive evaluation: 1) image denoising by SGN + long-short fusion by SGN (denoted as TP1); 2) image deblurring by DeblurGAN + long-short fusion by SGN (denoted as TP2); 3) long-short fusion by SGN + long-short fusion refinement by SGN (the same workflow as D2HNet, denoted as TP3). The same data processing schemes are applied to other methods.

We illustrate the generated samples on real photos in Figure 6. From image pairs ♯1 and ♯3, the black backgrounds of D2HNet results are cleaner than other

Fig. 6. Visual comparisons of the proposed D2HNet with other methods. More results on both real photos and validation images are in the supplementary material.

methods, e.g., obvious artifacts in results of DenseFuse, DeblurGAN, and TP1-TP3. It demonstrates that D2HNet has a better denoising ability for inputs. For image pair ♯2, D2HNet can generate a clean and sharp result from extreme blurry inputs, while maintaining the denoising ability of dark regions; however, there lie in artifacts in the dark regions of others. From image pairs ♯3 and ♯4, we can see D2HNet has better edge preservation ability compared with others, e.g., letters and Chinese characters are sharper and cleaner.

The quantitative analysis is concluded in Table 1. Compared with other single-phase methods, D2HNet obtains 0.80~1.77db PSNR gain on 1440p. It also outperforms the simple concatenated methods (TP1-TP3) on both 1440p and 2880p, which demonstrates that D2HNet is more robust to different input resolutions. Since there is no ground truth for real photos, we conduct a human perceptual study on the results generated from different methods and there are 10 observers. In each comparison, a user is presented with a pair of restored images side by side of a shuffled sequence. Then, the user chooses one result that produces cleaner and sharper images than others. The preference rates (PRs) are concluded in Table 2, where there are 79.28%~86.07% votes for D2HNet. The majority of users thought that D2HNet achieves higher image quality than compared methods. It demonstrates that D2HNet recovers images with better details and textures and well addresses the domain gap issue.

Table 3. Comparisons of D2HNet and other single image denoising methods.

Table 4. Comparisons of D2HNet and other single image deblurring methods.

Method	1440p val data		2880p val data	
	PSNR	SSIM	PSNR	SSIM
DnCNN [90]	32.20	0.9192	33.61	0.9265
MemNet [65]	33.74	0.9517	35.73	0.9644
MWCNN [43]	32.47	0.9372	34.71	0.9554
SGN [25]	33.94	0.9576	36.42	0.9713
RIDNet [2]	33.29	0.9462	35.55	0.9621
MIRNet [84]	33.98	0.9565	36.36	0.9708
REDI [38]	28.60	0.8964	31.54	0.9431
DeamNet [55]	33.78	0.9531	36.26	0.9685
MPRNet [85]	34.00	0.9568	36.25	0.9712
D2HNet	34.67	0.9639	36.85	0.9767

Method	1440p val data		2880p val data	
	PSNR	SSIM	PSNR	SSIM
DeepDeblur [51]	23.51	0.8252	23.80	0.8731
SRN [66]	23.99	0.8363	24.11	0.8780
DeblurGAN [36]	24.23	0.8399	24.13	0.8749
DeblurGANv2 [37]	23.88	0.8059	23.67	0.8359
DMPHN [87]	21.73	0.7807	22.38	0.8447
MPRNet [85]	22.97	0.8072	22.61	0.8438
HINet [12]	22.39	0.7586	21.93	0.7879
MIMOUNet [15]	21.11	0.7756	21.19	0.8355
MIMOUNet++ [15]	21.10	0.7753	21.25	0.8373
D2HNet	34.67	0.9639	36.85	0.9767

5.3   Single-image Denoising and Deblurring Method Experiments

We compare D2HNet and SOTA image denoising $[2, 25, 38, 43, 55, 65, 84, 85, 90]$ and deblurring $[12, 15, 36, 37, 51, 66, 85, 87]$ methods. Short-exposure images serve as inputs for denoising methods and s_{first} is ground truth. Long-exposure images serve as inputs for deblurring methods and l_{last} is ground truth.

We illustrate the generated samples on real photos in Figure 7. From ♯1 and ♯2, single image denoising methods cannot restore details of the roof (♯1) and the textures of curtains (♯2). However, D2HNet produces richer details since it fuses the information from the long-exposure input, where the textures are more distinguishable than the highly noisy short-exposure input. From ♯3 and ♯4, single image deblurring methods cannot recover either small blur or severe blur. The superiority of D2HNet comes from two reasons. On one hand, although the other methods estimate motion fields from a single long-exposure input, D2HNet utilizes the position information from the short-exposure input to guide the deblurring. On the other hand, a domain gap exists between training and testing data. Without proper handling, these methods degrade to mainly removing noises when encountering very large blurs in the testing images. Whereas, our architecture involves the DeblurNet which operates on a fixed resolution to better generalize on large blur. We also report the quantitative performance of all methods on the validation set in Table 3 and 4. Compared with single-image-based methods, D2HNet obtains giant increases on both metrics since it fuses more information from both long- and short-exposure inputs.

5.4   Ablation Study

We conduct the ablation study for the D2HNet, where the benchmark results are concluded in Table 5 and visual results are illustrated in Figure 8[1]. The analysis for different ablation study items is as follows:

Training Strategy. Dual inputs are significant for D2HNet to get more performance gain. Compared with only using long- or short-exposure input (settings 1)

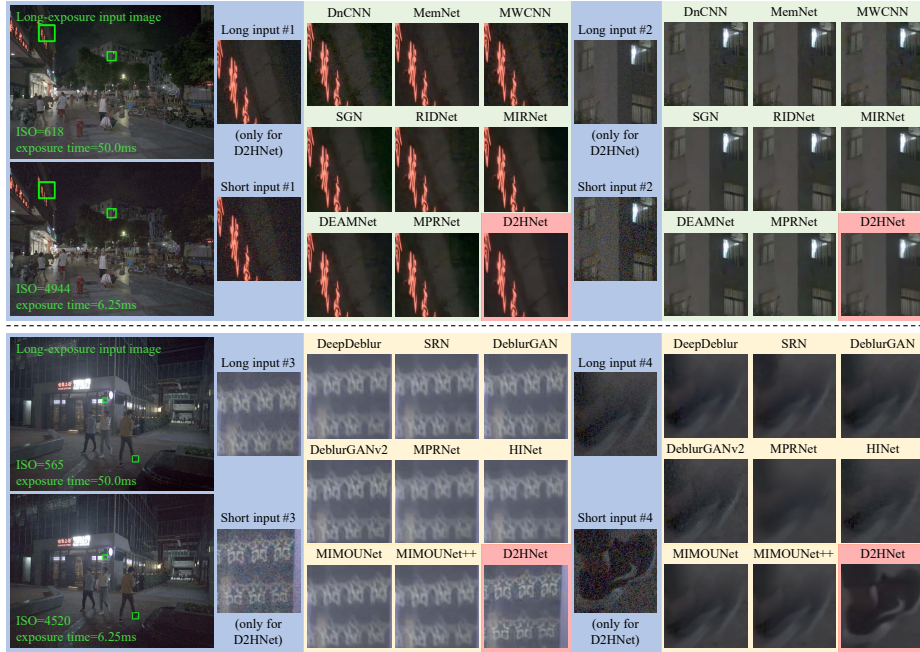[1] We thank Chao Wang in the SenseTime Research for helping capture the image.

Fig. 7. Visual comparisons of the proposed D2HNet with single image denoising methods (upper ♯1 and ♯2) and single image deblurring methods (lower ♯3 and ♯4).

and 2)), two inputs improve PSNR by 9.68dB and 0.59dB, respectively. Aligning the long-exposure input with short-exposure input (i.e., s_{first} as GT) also helps transfer textures from long-exposure input, which brings 4.51dB gain compared with l_{last} as GT (setting 3)). We can also see settings 1-3) cannot recover the details and remove artifacts (e.g., the face contour and eyes in ♯1).

Network Components. Alignment block makes the D2HNet better fuses features from the long-exposure input. In setting 4), we replace deformable convolutions with ordinary convolutions, forcing the network to apply rigid filters at all the spatial locations in the features, which brings a decrease of 0.24dB. In setting 5), we remove all Alignment and Feature fusion blocks, leading to a notable performance decrease of EnhanceNet (1.4dB) since the hierarchical information is excluded. From settings 6) and 7), the tail Residual block brings 0.5dB gain, while the full EnhanceNet brings 4.19dB gain since it learns rich textures and details. In addition, settings 4-7) produce blurry outputs and vague details (i.e., the billboard in ♯2), which show the importance of every component.

Data Processing Schemes. VarmapSelection balances the training data distribution, where D2HNet better generalizes to blurry or misaligned long-exposure inputs and learns to extract textures from them, e.g., D2HNet produces sharper results than setting 8) in ♯2. Illumination Adjustment generates more low-brightness training images, helping the D2HNet obtain better performance in dark regions.
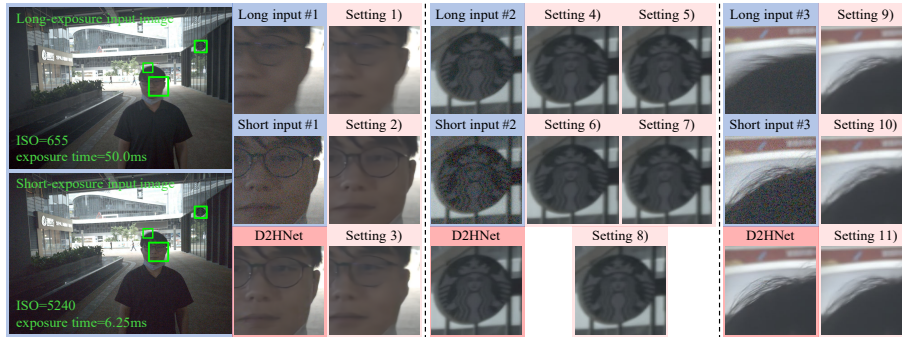
Fig. 8. Visual comparisons of D2HNet ablation study.

Table 5. Comparisons of D2HNet and ablation settings on 1440p validation data.

Ablation Study Setting	PSNR	SSIM	Ablation Study Setting	PSNR	SSIM
1) Only long input, l_{last} as ground truth	24.99	0.8610	7) w/o EnhanceNet (only DeblurNet)	30.48	0.9259
2) Only short input	34.08	0.9579	8) w/o VarmapSelection	34.24	0.9604
3) Long-short inputs, l_{last} as ground truth	30.16	0.9293	9) w/o Illumination Adjustment	34.31	0.9596
4) Replacing EnhanceNet Alignment block	34.43	0.9610	10) w/o Color Adjustment	34.58	0.9620
5) w/o EnhanceNet feature-level short-cuts	33.27	0.9530	11) w/o CutNoise	34.42	0.9616
6) w/o EnhanceNet tail Residual block	34.17	0.9602	D2HNet (full)	34.67	0.9639

Color Adjustment and CutNoise balance the usage of long-short inputs, encouraging sharper results. As shown in ♯3, if dropping each of them (settings 9-11)), the network cannot recover clear details of the black hair. Also, settings 8-11) result in the decreases of PSNR by 0.43dB, 0.36dB, 0.09dB, and 0.25dB, respectively. It is obvious that every data processing scheme is significant.

6 Conclusion

In this paper, we present a D2HNet framework for robust night image restoration based on long- and short-exposure inputs. It deblurs and restores sharp outputs from the long-exposure image under the guidance of the short-exposure image to obtain accurate colors, trivial noises, and sharp edges. It includes two sequential subnets: DeblurNet to remove blur on a fixed size and EnhanceNet to refine and sharpen the output of DeblurNet. For training, we synthesize the D2-Dataset including 6853 high-quality image tuples with multiple types and levels of blur. We propose a VarmapSelection scheme to generate highly blurry patches and assist the convergence of D2HNet. We also use a CutNoise scheme to enhance textures and details by enforcing D2HNet to learn how and where to deblur. For evaluation, we compare the proposed D2HNet with SOTA long-short fusion methods, and single image denoising and deblurring methods on the D2-Dataset validation set and real-world photos. The experimental results on both validation set and real-world photos show better performance achieved by the D2HNet.

References

1. Abdelhamed, A., Brubaker, M.A., Brown, M.S.: Noise flow: Noise modeling with conditional normalizing flows. In: Proc. ICCV. pp. 3165–3173 (2019)
2. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: Proc. ICCV. pp. 3155–3164 (2019)
3. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: Proc. CVPR. pp. 11036–11045 (2019)
4. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. CVPR. vol. 2, pp. 60–65 (2005)
5. Byun, J., Cha, S., Moon, T.: Fbi-denoiser: Fast blind image denoiser for poisson-gaussian noise. In: Proc. CVPR. pp. 5768–5777 (2021)
6. Chakrabarti, A.: A neural approach to blind motion deblurring. In: Proc. ECCV. pp. 221–235 (2016)
7. Chan, K.C., Wang, X., Yu, K., Dong, C., Loy, C.C.: Basicvsr: The search for essential components in video super-resolution and beyond. In: Proc. CVPR. pp. 4947–4956 (2021)
8. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In: Proc. CVPR. pp. 5972–5981 (2022)
9. Chang, M., Feng, H., Xu, Z., Li, Q.: Low-light image restoration with short-and long-exposure raw pairs. IEEE Trans. Multimedia **24**, 702–714 (2021)
10. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proc. CVPR. pp. 3291–3300 (2018)
11. Chen, J., Chen, J., Chao, H., Yang, M.: Image blind denoising with generative adversarial network based noise modeling. In: Proc. CVPR. pp. 3155–3164 (2018)
12. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: Hinet: Half instance normalization network for image restoration. In: Proc. CVPRW. pp. 182–192 (2021)
13. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1256–1272 (2016)
14. Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: Nbnet: Noise basis learning for image denoising with subspace projection. In: Proc. CVPR. pp. 4896–4906 (2021)
15. 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: Proc. ICCV. pp. 4641–4650 (2021)
16. Choi, B.D., Jung, S.W., Ko, S.J.: Motion-blur-free camera system splitting exposure time. IEEE Trans. Consum. Electron. **54**(3), 981–986 (2008)
17. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
18. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proc. ICCV. pp. 764–773 (2017)
19. Deng, J., Wang, L., Pu, S., Zhuo, C.: Spatio-temporal deformable convolution for compressed video quality enhancement. In: Proc. AAAI. vol. 34, pp. 10696–10703 (2020)
20. Dudhane, A., Zamir, S.W., Khan, S., Khan, F.S., Yang, M.H.: Burst image restoration and enhancement. In: Proc. CVPR. pp. 5759–5768 (2022)
21. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: Proc. CVPR. pp. 3848–3856 (2019)

22. Godard, C., Matzen, K., Uyttendaele, M.: Deep burst denoising. In: Proc. ECCV. pp. 538–554 (2018)
23. 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: Proc. CVPR. pp. 2319–2328 (2017)
24. Gu, C., Lu, X., He, Y., Zhang, C.: Blur removal via blurred-noisy image pair. IEEE Trans. Image Process. **30**, 345–359 (2020)
25. Gu, S., Li, Y., Gool, L.V., Timofte, R.: Self-guided network for fast image denoising. In: Proc. ICCV. pp. 2511–2520 (2019)
26. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proc. CVPR. pp. 1712–1722 (2019)
27. Guo, S., Yang, X., Ma, J., Ren, G., Zhang, L.: A differentiable two-stage alignment scheme for burst image reconstruction with large shift. In: Proc. CVPR. pp. 17472–17481 (2022)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR. pp. 770–778 (2016)
29. Hu, X., Ma, R., Liu, Z., Cai, Y., Zhao, X., Zhang, Y., Wang, H.: Pseudo 3d autocorrelation network for real image denoising. In: Proc. CVPR. pp. 16175–16184 (2021)
30. Ji, S.W., Lee, J., Kim, S.W., Hong, J.P., Baek, S.J., Jung, S.W., Ko, S.J.: Xydeblur: Divide and conquer for single image deblurring. In: Proc. CVPR. pp. 17421–17430 (2022)
31. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: Proc. CVPR. pp. 9000–9008 (2018)
32. Karadeniz, A.S., Erdem, E., Erdem, A.: Burst photography for learning to enhance extremely dark images. IEEE Trans. Image Process. **30**, 9372–9385 (2021)
33. Kim, Y., Soh, J.W., Park, G.Y., Cho, N.I.: Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In: Proc. CVPR. pp. 3482–3492 (2020)
34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. ICLR (2014)
35. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: Proc. NeurIPS. pp. 1033–1041 (2009)
36. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proc. CVPR. pp. 8183–8192 (2018)
37. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In: Proc. ICCV. pp. 8878–8887 (2019)
38. Lamba, M., Mitra, K.: Restoring extremely dark images in real time. In: Proc. CVPR. pp. 3487–3497 (2021)
39. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: Proc. CVPR. pp. 2657–2664 (2011)
40. Li, H., Wu, X.J.: Densefuse: A fusion approach to infrared and visible images. IEEE Trans. Image Process. **28**(5), 2614–2623 (2018)
41. Li, M., Liu, J., Yang, W., Sun, X., Guo, Z.: Structure-revealing low-light image enhancement via robust retinex model. IEEE Trans. Image Process. **27**(6), 2828–2841 (2018)
42. Liba, O., Murthy, K., Tsai, Y.T., Brooks, T., Xue, T., Karnad, N., He, Q., Barron, J.T., Sharlet, D., Geiss, R., et al.: Handheld mobile photography in very low light. ACM Trans. on Graphics **38**(6), 1–16 (2019)

43. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-cnn for image restoration. In: Proc. CVPRW. pp. 773–782 (2018)
44. Liu, W., Yan, Q., Zhao, Y.: Densely self-guided wavelet network for image denoising. In: Proc. CVPRW. pp. 432–433 (2020)
45. Liu, Y., Qin, Z., Anwar, S., Ji, P., Kim, D., Caldwell, S., Gedeon, T.: Invertible denoising network: A light solution for real noise removal. In: Proc. CVPR. pp. 13365–13374 (2021)
46. Liu, Z., Yuan, L., Tang, X., Uyttendaele, M., Sun, J.: Fast burst images denoising. ACM Trans. on Graphics **33**(6), 1–9 (2014)
47. Luo, Z., Yu, L., Mo, X., Li, Y., Jia, L., Fan, H., Sun, J., Liu, S.: Ebsr: Feature enhanced burst super-resolution with deformable alignment. In: Proc. CVPRW. pp. 471–478 (2021)
48. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Proc. NeurIPS. pp. 2802–2810 (2016)
49. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: Proc. CVPR. pp. 2502–2510 (2018)
50. Mustaniemi, J., Kannala, J., Matas, J., Särkkä, S., Heikkilä, J.: Lsd$_2$ - joint denoising and deblurring of short and long exposure images with convolutional neural networks. In: Proc. BMVC (2020)
51. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proc. CVPR. pp. 3883–3891 (2017)
52. Nimisha, T.M., Kumar Singh, A., Rajagopalan, A.N.: Blur-invariant deep learning for blind-deblurring. In: Proc. ICCV. pp. 4752–4760 (2017)
53. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: Proc. ECCV. pp. 327–343 (2020)
54. Purohit, K., Rajagopalan, A.: Region-adaptive dense network for efficient motion deblurring. In: Proc. AAAI. vol. 34, pp. 11882–11889 (2020)
55. Ren, C., He, X., Wang, C., Zhao, Z.: Adaptive consistency prior based deep network for image denoising. In: Proc. CVPR. pp. 8596–8606 (2021)
56. Richardson, W.H.: Bayesian-based iterative method of image restoration. JoSA **62**(1), 55–59 (1972)
57. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Proc. ECCV. pp. 184–201 (2020)
58. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proc. MICCAI. pp. 234–241 (2015)
59. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena **60**(1-4), 259–268 (1992)
60. Simoncelli, E.P., Adelson, E.H.: Noise removal via bayesian wavelet coring. In: Proc. ICIP. vol. 1, pp. 379–382 (1996)
61. Son, C.H., Choo, H., Park, H.M.: Image-pair-based deblurring with spatially varying norms and noisy image updating. Journal of Visual Comm. and Image Rep. **24**(8), 1303–1315 (2013)
62. Son, C.H., Park, H.M.: A pair of noisy/blurry patches-based psf estimation and channel-dependent deblurring. IEEE Trans. Consum. Electron. **57**(4), 1791–1799 (2011)
63. Suin, M., Purohit, K., Rajagopalan, A.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: Proc. CVPR. pp. 3606–3615 (2020)
64. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proc. CVPR. pp. 769–777 (2015)

65. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: Proc. ICCV. pp. 4539–4547 (2017)
66. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: Proc. CVPR. pp. 8174–8182 (2018)
67. Tian, Y., Zhang, Y., Fu, Y., Xu, C.: Tdan: Temporally-deformable alignment network for video super-resolution. In: Proc. CVPR. pp. 3360–3369 (2020)
68. Tico, M., Gelfand, N., Pulli, K.: Motion-blur-free exposure fusion. In: Proc. ICIP. pp. 3321–3324 (2010)
69. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: Proc. CVPRW. pp. 1–10 (2019)
70. Wang, Y., Cao, Y., Zha, Z.J., Zhang, J., Xiong, Z., Zhang, W., Wu, F.: Progressive retinex: Mutually reinforced illumination-noise perception network for low-light image enhancement. In: Proc. ACM MM. pp. 2015–2023 (2019)
71. Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., Wang, J.: Practical deep raw image denoising on mobile devices. In: Proc. ECCV. pp. 1–16 (2020)
72. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
73. Wei, K., Fu, Y., Yang, J., Huang, H.: A physics-based noise formation model for extreme low-light raw denoising. In: Proc. CVPR. pp. 2758–2767 (2020)
74. Whang, J., Delbracio, M., Talebi, H., Saharia, C., Dimakis, A.G., Milanfar, P.: Deblurring via stochastic refinement. In: Proc. CVPR. pp. 16293–16303 (2022)
75. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. Int. J. Comput. Vis. **98**(2), 168–186 (2012)
76. Xia, Z., Perazzi, F., Gharbi, M., Sunkavalli, K., Chakrabarti, A.: Basis prediction networks for effective burst denoising with large kernels. In: Proc. CVPR. pp. 11844–11853 (2020)
77. Xing, Y., Qian, Z., Chen, Q.: Invertible image signal processing. In: Proc. CVPR. pp. 6287–6296 (2021)
78. Xu, X., Li, M., Sun, W.: Learning deformable kernels for image and video denoising. arXiv preprint arXiv:1904.06903 (2019)
79. Yoo, J., Ahn, N., Sohn, K.A.: Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In: Proc. CVPR. pp. 8375–8384 (2020)
80. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. ACM Trans. on Graphics **26**(3), 1–es (2007)
81. Yuan, Y., Su, W., Ma, D.: Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In: Proc. CVPR. pp. 3555–3564 (2020)
82. Yue, Z., Yong, H., Zhao, Q., Meng, D., Zhang, L.: Variational denoising network: Toward blind noise modeling and removal. Proc. NeurIPS **32**, 1690–1701 (2019)
83. 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: Proc. CVPR. pp. 2696–2705 (2020)
84. 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: Proc. ECCV. pp. 492–511 (2020)
85. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proc. CVPR. pp. 14821–14831 (2021)

86. Zhang, B., Jin, S., Xia, Y., Huang, Y., Xiong, Z.: Attention mechanism enhanced kernel prediction networks for denoising of burst images. In: Proc. ICASSP. pp. 2083–2087 (2020)
87. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: Proc. CVPR. pp. 5978–5986 (2019)
88. 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: Proc. CVPR. pp. 2521–2529 (2018)
89. Zhang, J., Cao, Y., Fang, S., Kang, Y., Wen Chen, C.: Fast haze removal for nighttime image using maximum reflectance prior. In: Proc. CVPR. pp. 7418–7426 (2017)
90. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017)
91. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Trans. Image Process. **27**(9), 4608–4622 (2018)
92. Zhang, Y., Wang, C., Maybank, S.J., Tao, D.: Exposure trajectory recovery from motion blur. IEEE Trans. Pattern Anal. Mach. Intell. (2021)
93. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **43**(7), 2480–2495 (2020)
94. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proc. CVPR. pp. 9308–9316 (2019)