

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Digital Object Identifier 10.1109/TIP.2024.3515873

# Modeling Dual-Exposure Quad-Bayer Patterns for Joint Denoising and Deblurring

Yuzhi Zhao, *Graduate Student Member, IEEE*, Lai-Man Po, *Senior Member, IEEE*,  
 Xin Ye, Yongzhe Xu, Qiong Yan

**Abstract**—Image degradation caused by noise and blur remains a persistent challenge in imaging systems, stemming from limitations in both hardware and methodology. Single-image solutions face an inherent tradeoff between noise reduction and motion blur. While short exposures can capture clear motion, they suffer from noise amplification. Long exposures reduce noise but introduce blur. Learning-based single-image enhancers tend to be over-smooth due to the limited information. Multi-image solutions using burst mode avoid this tradeoff by capturing more spatial-temporal information but often struggle with misalignment from camera/scene motion. To address these limitations, we propose a physical-model-based image restoration approach leveraging a novel dual-exposure Quad-Bayer pattern sensor. By capturing pairs of short and long exposures at the same starting point but with varying durations, this method integrates complementary noise-blur information within a single image. We further introduce a Quad-Bayer synthesis method (B2QB) to simulate sensor data from Bayer patterns to facilitate training. Based on this dual-exposure sensor model, we design a hierarchical convolutional neural network called QRNet to recover high-quality RGB images. The network incorporates input enhancement blocks and multi-level feature extraction to improve restoration quality. Experiments demonstrate superior performance over state-of-the-art deblurring and denoising methods on both synthetic and real-world datasets. The code, model, and datasets are publicly available at <https://github.com/zhaoyuzhi/QRNet>.

**Index Terms**—Image Denoising, Image Deblurring, Quad-Bayer Sensor, Neural Networks.

## I. INTRODUCTION

MODERN imaging pipelines inherently suffer from noise and blur artifacts stemming from fundamental hardware constraints, including shot noise, read noise, limited sensor resolution, quantization error, and suboptimal capture conditions such as low light. Additionally, motion blur remains problematic, particularly with handheld cameras. Noise and blur reduction have thus represented long-standing challenges in computational photography. With the proliferation of digital photography, producing high-quality images under real-world conditions is an important need. The straightforward approach of extending exposures gathers more photons but requires static scenes, often impractical outside controlled settings. Effective post-processing methods for noise and blur are thus essential.

Manuscript received October 14, 2022, revised September 21, 2023, accepted December 3, 2024. (*Corresponding author: Yuzhi Zhao*).

Y. Zhao, L.-M. Po are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China (e-mail: yzzhao2-c@my.cityu.edu.hk; eelmpo@cityu.edu.hk).

X. Ye, Y. Xu, Q. Yan are with the SenseTime Research and Tetras.AI, China (e-mail: yexinzju@gmail.com; xuyongzhe94@gmail.com; sophie.yanqiong@gmail.com).

Denoising typically relies on accurate noise models, often trained on RGB data despite noise arising in raw sensor data. Recent works have aimed to simulate raw noise more effectively [1], [2] or improve noise model estimation [3]–[7], but texture preservation remains lacking. For deblurring, the common blind setting of unknown kernel or motion complicates modeling, while domain gaps between synthetic and real blur persist [8]–[10]. Joint denoising-deblurring approaches [11]–[13] using short- and long-exposure burst images provide more signal, but unaligned inputs make fusion difficult.

The characteristics of short- and long-exposure images exhibit a complementary nature, with short-exposure pixels capturing sharp edges and long-exposure pixels containing minimal noise and accurate colors. By merging these two exposures, we can effectively restore textures while accurately estimating blur kernels simultaneously. A recent advancement in the field of image processing is the introduction of a sensor known as Quad-Bayer [14], [15], which incorporates two types of exposures within a single mosaic pattern. This pattern, serving as the minimum unit of the color filter array (CFA), possesses a resolution of  $4 \times 4$ , as depicted in Figure 1 (i). To fully leverage the capabilities of the Quad-Bayer sensor, we propose the development of a joint pipeline for image deblurring and denoising. Our proposed solution offers two significant advantages:

- 1) The utilization of pixels from both short and long exposures simultaneously enhances the image reconstruction task by incorporating a larger amount of information compared to single-image methods [1]–[6], [8]–[10].
- 2) The synchronization of pixels from both exposure types, as illustrated in Figure 1 (j), eliminates any misalignment concerns typically encountered in burst-image methods [11]–[13].

We propose QRNet, an end-to-end neural network for joint deblurring and denoising of Quad-Bayer RAW images. To accommodate the Quad-Bayer layout, we employ the Pixel Unshuffle technique [16] to reorganize the colors and exposures for the single-channel input, resulting in 16 channels. To mitigate the presence of discontinuous artifacts caused by downsampling, we introduce an input augmentation block (IEB). This block incorporates three additional branches, obtained through average pooling and convolutions, alongside the 16 channels obtained via Pixel Shuffle. Subsequently, all branches are merged using a channel attention scheme. Furthermore, QRNet is structured as a multi-level architecture, facilitating feature interactions between levels and enhancing

the network's ability to represent features effectively.

To facilitate the training and evaluation of our proposed imaging pipeline, we have introduced a new synthetic dataset called the Quad-Bayer to RGB dataset (QR dataset). This dataset is essential as it is challenging to amass a large collection of Quad-Bayer and RGB image pairs using a camera equipped with a Quad-Bayer sensor that allows for flexible adjustment of camera parameters such as f-number, ISO, and focal length. The images in the dataset were originally captured using a DSLR camera (Nikon D5200) and encompass diverse scenes, including urban environments, parks, forests, mountains, and more. To generate the synthetic Quad-Bayer and RGB image pairs, we sample Quad-Bayer counterparts from pairs of long- and short-exposure Bayer RAW images. To ensure proper alignment between the synthetic Quad-Bayer and RGB image pairs, we propose a Bayer to Quad-Bayer (B2QB) sampling scheme. The resulting dataset comprises 701 training pairs and 30 validation pairs, all with a resolution of  $3016 \times 2008$  pixels.

We conducted a comprehensive evaluation of QRNet using both the QR dataset and real Quad-Bayer data. Both qualitative and quantitative experiments have shown that the proposed pipeline is capable of restoring images with intense noises or extremely blurry scenes, which outperforms existing approaches. Overall, the main contributions of this paper are as follows:

- 1) We propose a novel solution that tackles the challenges of denoising and deblurring using Quad-Bayer sensors in a joint manner;
- 2) We introduce a Bayer to Quad-Bayer sampling method (B2QB) and curate a substantial QR dataset comprising 731 aligned Quad-Bayer and RGB image pairs;
- 3) We conduct extensive experiments, comparing image denoising and deblurring methods in both RGB space and RAW space, to demonstrate the significant advancements achieved by QRNet.

In the following, we will review some more background and previous works in Section II, then introduce our method and the QR dataset in Section III and IV respectively, followed by our detailed experiment and analysis in Section V. Finally, we conclude the paper in Section VI.

## II. RELATED WORK

**Color Filter Array (CFA) and Quad-Bayer Pattern.** Image sensors are widely applied in modern color digital cameras. Considering the restriction on camera sizes, especially for mobile phone cameras, portable image sensors are in great demand. Color filters, as a significant component of image sensors, are designed to reduce the size and cost. These color filters are arranged in a mosaic pattern and each filter allows only one of the primary colors (R, G, B) to pass through. The arrangement is called a color filter array (CFA). For each pixel, only one color channel, either R, G, or B, is recorded. The missing ones are reconstructed by demosaicking. The first CFA was proposed by Bayer [17], as shown in Figure 1 (a). The minimum unit of the Bayer pattern is a  $2 \times 2$  sub-block, where the green channel is sampled at every other pixel and

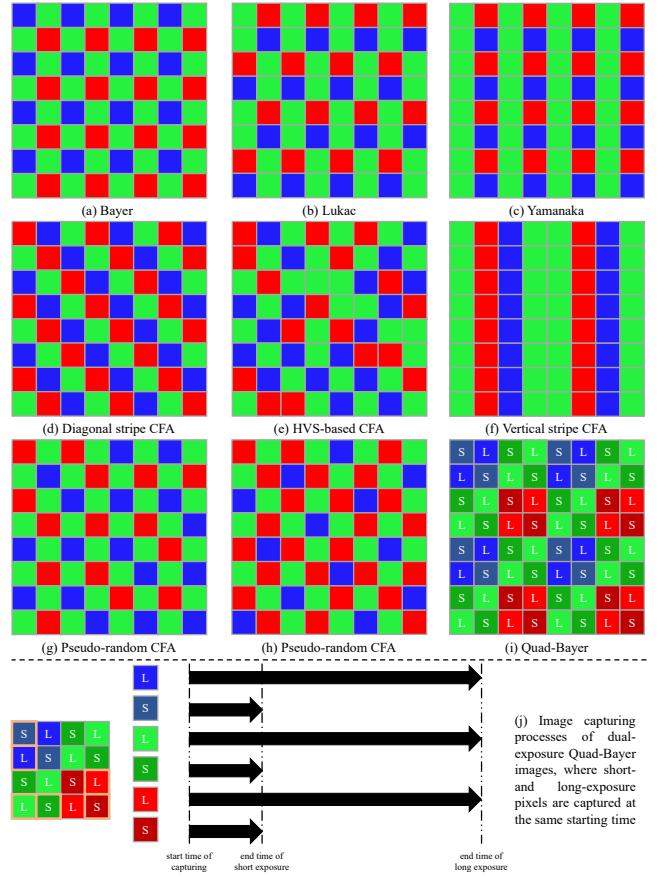


Fig. 1. Illustration of (a) Normal Bayer pattern, (b-h) Other Bayer patterns with single exposure, (i) Quad-Bayer pattern with both long and short exposures, and (j) Image capturing process of dual-exposure Quad-Bayer images.

red/blue channels are sampled once. Later on, many other sampling configurations have been proposed such as Lukac CFA [18], Yamanaka CFA [19], and HVS-based CFA [20]. They are shown in Figure 1 (b), (c), and (e), respectively. Though some of them improve the reconstruction accuracy or minimize an error, the sensor only records pixels under a single exposure. Recently, a novel sensor pixel layout called Quad-Bayer was introduced by Sony. The Quad-Bayer filter is an extension of the classic Bayer pattern, as shown in Figure 1 (i). The structure enables two exposures within a group of four pixels thus avoiding multi-frame photography. It also alleviates the ghosting issue compared with classical multi-frame photography.

**Image Deblurring.** The target of image deblurring is estimating the blur kernel in the form of a convolution kernel or a motion field and then reconstructing the clear image. Deblurring methods can be categorized into blind and non-blind (i.e., blur convolution kernel or a motion field is known). Most of the classical methods addressed the non-blind deblurring issue by assuming blur kernel statistics and performing deconvolution. For instance, the Lucy-Richardson algorithm [21] assumed a point spread function and the observed image should follow a Poisson distribution. However, it suffers from ringing artifacts. To address that issue, some methods used additional priors as

regularizations such as GMM [22], patch [23], sparsity [24], and  $l_0$  regularization [25].

Since neural networks can extract rich data prior from large numbers of training pairs, they have been widely used to address the non-blind deblurring problem. From the perspective of network architecture, Sun *et al.* [26] first proposed a multi-layer network to estimate blur kernel. Gong *et al.* [27] utilized a fully convolutional network to produce a pixel-wise motion flow map. The output images are obtained by performing deconvolution for both of them. Nah *et al.* [28] proposed an end-to-end non-blind deblurring method, which has three levels with different input resolutions. It was enhanced by SRN [29], DMPHN [30], and MPRNet [31] through a similar multi-scale training strategy. Furthermore, recent works have combined other generative models such as RNN-Deblur [32], Flow-guided Deblur [33], Multi-stage Deblurring [34], and Diffusion-guided Deblurring [35]. From the perspective of training strategy, DeblurGAN [8] and DeblurGANv2 [9] introduced adversarial training [36] to existing architectures. More recently, some works have developed self-supervised training strategies such as Reblur2Deblur [37], Class-Specific Deblurring [38], Deblurring by Reblurring [39], SelfDeblur [40], and Meta-Auxiliary Learning [41].

**Image Denoising.** The image denoising aims to remove noises and artifacts in an image. It also can be categorized into blind and non-blind denoising (i.e., noise model or noise kernel is known). Traditional non-blind image denoising methods often assumed a noise model first and then designed effective filters, e.g., diffusion [42], total variation [43], wavelet coring [44], non-local means [45], Wiener filter [46], sparse coding [47]–[49] block-matching and 3-D filtering (BM3D) [50]. The noise model was often assumed to be additive white Gaussian noise (AWGN) or Poisson-Gaussian model (G-P). Though these approaches achieved good results on such specific noise patterns, they are not robust to real noises and have relatively high computational complexity.

To address the blind denoising problem, recent works often used neural networks including [31], [51]–[63]. Due to the strong fitting capability of neural networks, blind denoising can be conducted during inference following extensive training on real-world noisy-clean image pairs. As for the network design, researches have considered U-Net [64], residual blocks [65], attention blocks [59], flow model [66], and diffusion model [67]. Furthermore, to estimate the noise level for training, CBDNet [68], VDN [69], and AINDNet [70] combine a denoising network and a noise estimation network. Some works [71]–[77] have performed denoising in an unsupervised manner without any training data or explicit noise models.

**Image Restoration with Burst Long- and Short-images.** Though traditional learning-based single-image deblurring and denoising methods can be potential solutions to remove artifacts, achieving practical performance is still challenging. To address this, some works proposed to use additional information, i.e., burst long- and short-exposure images. Yuan *et al.* [79] first used short-exposure images to guide the estimation of blur kernels, which are then used to restore blurry images. This method was improved by the non-uniform model [80] and neural networks [81]. However, their performance is restricted

TABLE I  
LIST OF ALL THE NOTATIONS USED IN THE PAPER.

Notation	Definition
$i$	Clean and sharp image (target)
$o$	Blurry and noisy observation image
$k$	Blur kernel imposed on the image
$n$	Additive noise imposed on the image
$n_s$	Shot noise, following Poisson distribution $\mathcal{P}$
$n_r$	Read noise, following Gaussian distribution $\mathcal{G}$
$n_q$	Quantization noise, following Uniform distribution $\mathcal{U}$
$K$	Noise parameter of shot noise $n_s$
$\sigma$	Noise parameter of read noise $n_r$
$qu$	Noise parameter of quantization noise $n_q$
$w$	White balance parameter for color channels
$\lambda$	Gamma correction parameter
$mov$	Moving image captured from “moving” clip for data synthesis
$sta$	Static image captured from “static” clip for data synthesis
$A$	Exposure ratio of long- and short-exposure pixels in a Quad-Bayer data
$x$	Synthetic clean dual-exposure Quad-Bayer data of QR dataset
$x_s$	Synthetic clean short-exposure Quad-Bayer data of QR dataset
$x_l$	Synthetic clean long-exposure Quad-Bayer data of QR dataset
$y$	Synthetic clean and sharp RGB image of QR dataset
$y_s$	Synthetic clean short-exposure RGB image of QR dataset, corresponding to $x_s$
$y_l$	Synthetic clean long-exposure RGB image of QR dataset, corresponding to $x_l$
$q$	Degraded Quad-Bayer data (blurry and noisy), processed from $x$
$G$	The proposed QRNet
$F$	Fast Fourier Transform (FFT) operator
$\alpha$	Frequency loss parameter
$\beta_1, \beta_2$	Adam optimizer [78] parameters

since the information of both inputs is not well fused. To achieve this goal, LSD2 [11] adopted a U-Net [64] and used the concatenated long- and short-exposure images as the input; but it did not consider the misalignment issue between them. Chang *et al.* [12] used a deformable convolution [82] since it explicitly aligned long- and short-exposure inputs. Shen *et al.* extended this idea in video restoration [83], where optical flows were used to align both inputs. Since the motion blur scales are in a wide range for real-world photos, those methods produce visual artifacts due to the domain gap between training samples and real-world data. Also, there is a readout delay between long- and short-exposure images during capturing. To address the problems, Zhao *et al.* proposed a two-stage D2HNet [13], where the input resolution of the first stage is fixed to perform accurate deblurring. Note that, there is no readout delay problem since long- and short-exposure pixels are recorded in one image, as shown in Figure 1 (j). The two kinds of pixels are compactly arranged so the misalignment is a controlled problem, i.e., the motion blur scales for long-exposure pixels can be estimated using neighboring short-exposure pixels. Therefore, we design a much faster approach for restoring Quad-Bayer sensor data based on these observations.

**RAW Image Restoration and RAW to RGB Mapping.** RAW is the direct output format of the camera, which contains richer information than RGB since there is no other image processing applied to it (e.g., image denoising, quantization, white balance, color correction, tone mapping, demosaicking, etc.). Thanks to the great fitting ability of neural networks, it

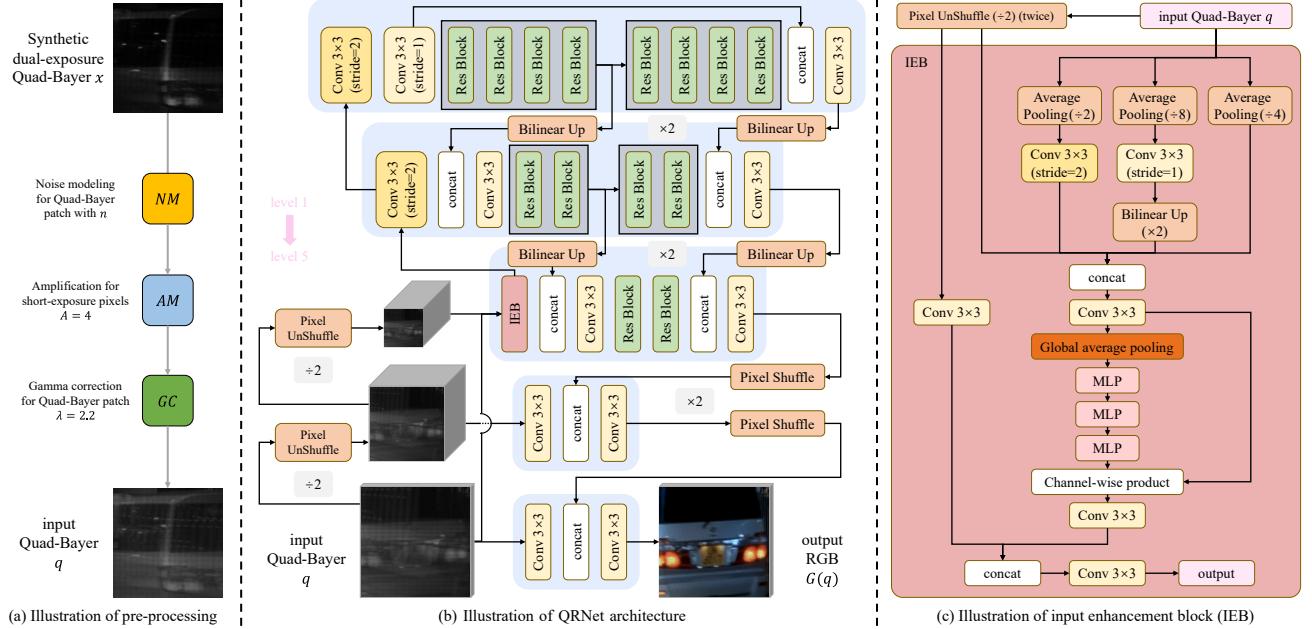


Fig. 2. Illustration of (a) Quad-Bayer pre-processing, (b) QRNet architecture (level 1 to 5 from top to bottom), and (c) input enhancement block (IEB).

is practical and more accurate to perform image denoising or deblurring on RAW data [84], [85]. Furthermore, researchers have developed joint image denoising and demosaicking models [86]–[89] by directly restoring a clean RGB image from a noisy RAW image. We observe that there is no explicit demosaicking method for Quad-Bayer RAW image and neural networks have the potential to map RAW to RGB. In this paper, we expand upon these applications by introducing a novel approach for jointly addressing image deblurring and denoising using a dual-exposure RAW image captured with the Quad-Bayer pattern.

### III. METHODOLOGY

#### A. Problem Formulation

We aim to recover a sharp and clean RGB image from a degraded dual-exposure RAW image with the Quad-Bayer pattern (hereinafter referred to as Quad-Bayer image). We first introduce an image restoration model. Suppose that  $i$  is a clean and sharp image and  $o$  is a blurry and noisy observation image. Assume that  $k$  is a blur kernel defined by some motion fields and  $n$  is an additive noise, the formation of degraded image  $o$  is:

$$o = k * i + n, \quad (1)$$

where  $*$  denotes the convolution operator. In addition, we want the model can also demosaic input Quad-Bayer data to RGB format. Assuming that  $q$  is a degraded Quad-Bayer image, we aim to recover a sharp and clean RGB image  $y$  by the proposed QRNet. We formulate the problem as maximizing a posteriori of the output image conditioned on input  $q$  and QRNet parameters  $\Theta$ :

$$\Theta^* = \arg \max_{\Theta} p(y|q, \Theta), \quad (2)$$

where  $y$  serves as the ground truth in the optimization. All the notations are concluded in Table I. Details on the training process are presented below.

#### B. Quad-Bayer Pre-processing

There are two degradation types (blur and noise) for real dual-exposure RAW images, and we need to model them when training the QRNet. Since we collect the synthetic training Quad-Bayer data  $x$  by capturing moving videos, the motion blur was modeled. To model RAW noises, we add noises to  $x$  based on [4]. In addition, we propose an amplification scheme and use the gamma correction to match the intensities between Quad-Bayer input and RGB ground truth [84], which contributes to the convergence of the QRNet. They are sequential operations as shown in Figure 2 (a) and the details are concluded as:

- 1) Noise modeling ( $NM$ ): Since the noises emerge on RAW images, we add noise  $n$  to the Quad-Bayer patch  $x$  based on the physical model [4], i.e.,  $NM(x) = x + n$ ;
- 2) Amplification ( $AM$ ): Since the exposure times of long- and short-exposure pixels are different, we multiply short-exposure pixels with a factor  $A$  to align their illuminations, i.e.,  $AM(x_{short\_pixels}) = A \cdot x_{short\_pixels}$ ;
- 3) Gamma correction ( $GC$ ): We apply a gamma correction to both long- and short-exposure pixels, which makes the overall illumination of Quad-Bayer images comparable with target RGB images. It is implemented by  $GC(x) = (x + \epsilon)^{(1/\lambda)}$ , where the factor  $\lambda = 2.2$  and  $\epsilon = 10^{-8}$  ( $\epsilon$  circumvents the numerical instability).

Note that the Quad-Bayer images are normalized to range of  $[0, 1]$  before performing pre-processing.

### C. Network Architecture

The proposed Quad-Bayer to RGB mapping network (QR-Net) is shown in Figure 2 (b). It is an end-to-end network that recovers an RGB image from an input Quad-Bayer image. It is also a multi-level architecture, and we define levels 1 to 5 from top to bottom based on the feature or image resolution, e.g., level 5 has the original resolution. We describe the designs in detail as follows:

**Data Re-arrangement.** Taking into account the Quad-Bayer layout, the short- and long-exposure pixels are densely packed within a single channel. Directly employing convolutional layers on this layout poses challenges for feature extraction, as the layers must additionally discern the spatial relationships between pixels captured at two different exposure levels. To mitigate this issue, we perform the Pixel Unshuffle operation [16] twice on the input Quad-Bayer data. This operation segregates pixels based on their exposure times and colors into distinct channels, as illustrated in Figure 3.

**Input enhancement block.** We notice that neighboring pixels of Pixel Unshuffle downsampled features are not continuous (e.g., pixels  $S_1^1$  and  $S_2^1$  in Figure 3). This sampling method might cause jagged artifacts. To alleviate it, we propose an input enhancement block (IEB), as shown in Figure 2 (c). In addition to Pixel Unshuffle operations, we include three more average pooling branches. It merges the intensity of neighboring pixels of original Quad-Bayer data that has better continuity. Then, we merge the three average pooling downsampled data and Pixel Unshuffle downsampled data by channel attention, inspired by [90]. After that, we combine the channel attention features with features extracted from Pixel Unshuffle downsampled data by another convolutional layer.

**Multi-level feature extraction.** Based on the result from IEB, we further define a three-level structure (upper three levels of Figure 2 (b)) to fuse the information from both short- and long-exposure pixels. We use residual blocks [65] at levels 1 to 3 to improve the feature representation. For levels 2 and 3, the information from the previous level is combined before entering the residual blocks. All the convolutional layers adopt  $3 \times 3$  kernels. Since the majority of operations are performed at levels 1 to 3, which have small image or feature resolutions, the architecture is efficient at inference.

**Feature interactions between levels.** Levels 1 to 5 of QRNet are related by upsampling the features from the previous level. For levels 1 to 3, there are two connections between every two neighboring levels at the start and the tail, respectively. The connection at the start combines the information from the previous level and the input of the next level. Then, we use a convolutional layer to emphasize the feature fusion and suppress the number of channels. It makes the current level possess multi-scale features, which enlarges the perceptive field and enhances the feature representation. The tail connection passes the whole information from the previous level to the next level. For levels 4 and 5, the relation is implemented by Pixel Shuffle upsampling and concatenation, which corresponds to the downsampling operations of the input pyramid. It also improves the image restoration results since every level can exploit multi-scale features.

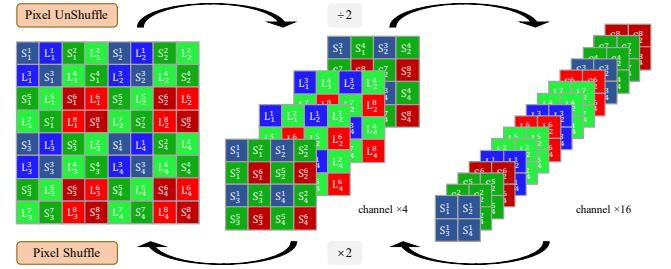


Fig. 3. Illustration of Pixel Unshuffle and Pixel Shuffle operations [16] for Quad-Bayer images, which are invertible.

### D. Loss Functions

Likewise existing image restoration methods, we exploit the L1 loss as reconstruction loss to make the generated image close to ground truth. It is defined as:

$$L_{recon} = \mathbb{E}[||\mathbf{G}(q) - y||_1], \quad (3)$$

where  $\mathbf{G}(q)$  and  $y$  represent the QRNet output and the ground truth RGB image, respectively.

To further reconstruct high-frequency parts of an image, we use a frequency loss [10] at the training. It measures the L1 distances between the generated image and the ground truth in the frequency domain as follows:

$$L_{freq} = \mathbb{E}[||\mathbf{F}(\mathbf{G}(q)) - \mathbf{F}(y)||_1], \quad (4)$$

where  $\mathbf{F}$  is the fast Fourier transform (FFT) operator that transforms the image from the spatial domain into the frequency domain. Combining the two loss functions, we define the overall loss as:

$$L_{overall} = L_{recon} + \alpha L_{freq}, \quad (5)$$

where  $\alpha$  is a trade-off parameter.

## IV. DATASET COLLECTION

In this section, we introduce the data collection details of the Quad-Bayer to RGB mapping dataset (QR dataset). The motivation that we synthesize training pairs instead of collecting them by Quad-Bayer module is in three folds:

- 1) It is tedious and time-consuming to change the settings for Quad-Bayer modules;
- 2) There are no effective demosaicking methods for Quad-Bayer sensor data (i.e., hard to obtain corresponding RGB images as the target for the proposed algorithm);
- 3) A perfect alignment is hard to obtain when capturing multiple Quad-Bayer and RGB pairs.

Therefore, we propose a data collection pipeline to create aligned and paired Quad-Bayer and RGB images. It includes four sequential steps:

**Preparation.** Considering the motivation, to build a sufficient dataset to learn the Quad-Bayer to RGB mapping, we capture Bayer RAW images first and then sample pixels to obtain their Quad-Bayer counterparts (i.e., B2QB scheme). Since Quad-Bayer RAW data includes two exposures, we need the paired long- and short-exposure Bayer RAW images. To capture these images, we first download 30 HD videos (2160p)

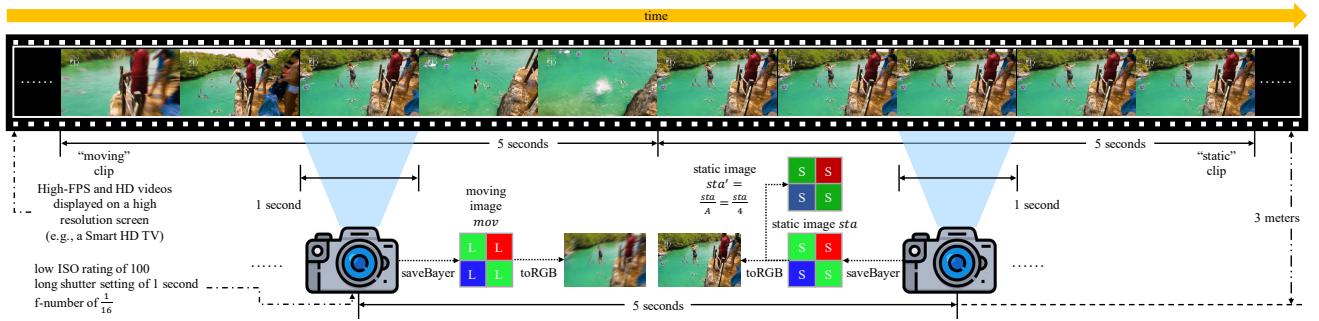


Fig. 4. Illustration of the data capturing workflow for a moving image and a static image from an edited video.

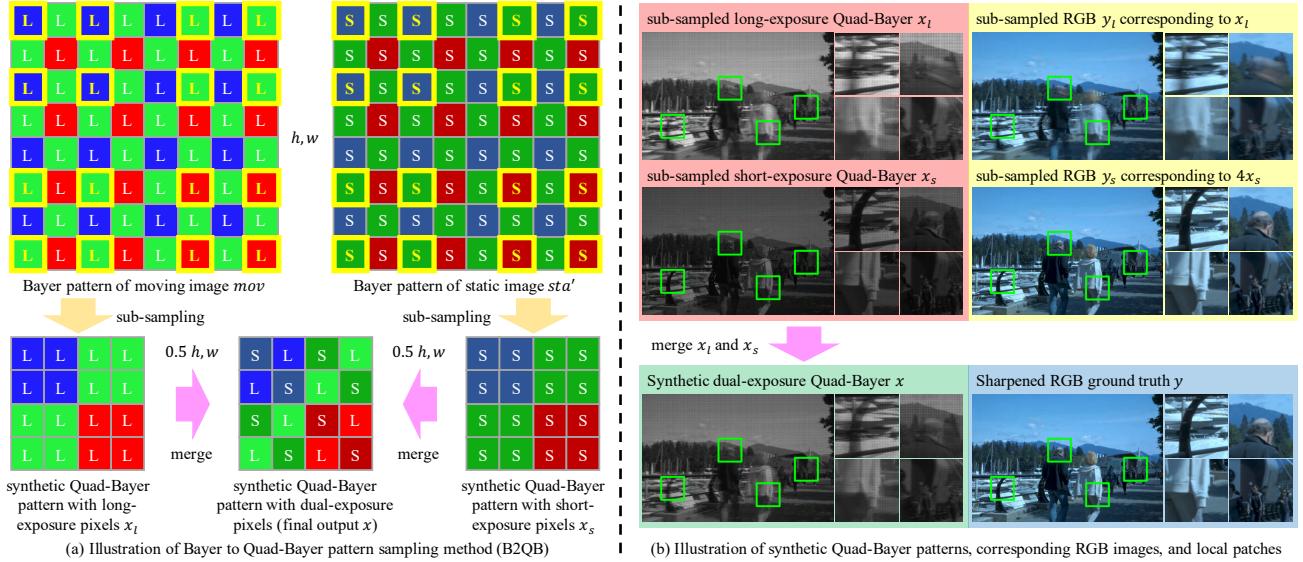


Fig. 5. Illustration of (a) Bayer to Quad-Bayer pattern sampling method (B2QB); (b) synthetic Quad-Bayer patterns, their corresponding RGB images, and four local patches extracted from the full resolution images. To better illustrate those images, we post-process them to have the same illuminations.

from YouTube, which cover different real-world scenes (e.g., country, city, mountain, forest, etc.) and are almost noise-free.

**Data modeling.** To better capture images, we post-process these videos to paired “moving” and “static” clips, for capturing long- and short-exposure images, respectively. In detail, we cut every original video into clips of 2.5 seconds with a sampling step of 45 seconds. For each clip, we interpolate it 2 times longer, i.e., 5 seconds. It is called a “moving” clip. From the “moving” clip, the middle frame is extracted and repeated to form a “static” clip of 5 seconds and the same FPS as the corresponding “moving” clip. These paired “moving” and “static” clips form a group and are composed into one single video for capturing, as shown in Figure 4.

**Data capturing.** To control the environment, we capture the edited clips on a high-resolution monitor with a DSLR camera, Nikon D5200, which captures RAW images with 14 bits. All other lighting sources are closed except the monitor. The display fully occupies the camera’s FOV. We use a low ISO rating of 100, a shutter setting of 1 second, and an f-number of  $\frac{1}{16}$  to get rid of noises as many as possible. A long shutter speed will induce overexposure and extreme blur and a short shutter speed will result in gloomy and noisy images, while a 1-second shutter speed is a trade-off setting. The RAW

images taken under such conditions are almost noise-free. The camera is placed 3 meters away from the monitor with a tripod and takes an image every 5 seconds controlled by a programmable shutter release. The workflow of capturing a pair of moving image  $mov$  and static image  $sta$  is illustrated in Figure 4. Since the overall exposure times and ISO ratings for  $mov$  and  $sta$  are the same, their brightness of them is the same. However, long- and short-exposure pixels of real Quad-Bayer images should have different brightness due to different exposure times. Therefore, we apply an exposure ratio  $A$  to the static image  $sta$ , i.e., the processed  $sta' = \frac{sta}{A}$ . In this paper, we set  $A = 4$  according to the real Quad-Bayer device.

**B2QB Scheme.** After capturing, we obtain pairs of blurry and sharp images with trivial noises and different brightness, i.e.,  $mov$  and  $sta'$ . They are regarded as long- and short-exposure images, respectively. Then, we synthesize the corresponding *Quad-Bayer data* following the B2QB scheme as illustrated in Figure 5 (a), which yields downsized images with half resolution of DSLR outputs, denoted as  $x_l$  and  $x_s$ . Finally, we combine the long- and short-exposure Quad-Bayer data to obtain the dual-exposure Quad-Bayer data  $x$ . For *target RGB images*, we propose to demosaic the Bayer patterns  $sta$  and apply the same sampling scheme in order to minimize

TABLE II

THE COMPARISONS OF QRNET AND OTHER METHODS BASED ON DUAL-EXPOSURE QUAD-BAYER IMAGES. DIFFERENT ROWS INCLUDE THE RESULTS FROM DIFFERENT METHODS AND THE LAST ROW IS THE PROPOSED QRNET. COLUMNS 1-4 INCLUDE THE METHODS, INPUT DATA STRUCTURE, MACS (ON  $512 \times 512$  RESOLUTION), AND RUNNING TIME (SECOND PER  $512 \times 512$  PATCH), RESPECTIVELY. COLUMNS 5-7 PSNR & SSIM RESULTS OF DIFFERENT NOISE PARAMETERS. THE RED, BLUE, AND GREEN COLORS DENOTE THE BEST, THE SECOND HIGHEST AND THE THIRD HIGHEST RESULTS, RESPECTIVELY.

Method	Input data structure	MACs ( $512 \times 512$ )	Running time (sec / $512 \times 512$ )	$K = 0.25, \sigma = 5$ PSNR	$K = 0.25, \sigma = 5$ SSIM	$K = 0.5, \sigma = 5$ PSNR	$K = 0.5, \sigma = 5$ SSIM	$K = 0.75, \sigma = 5$ PSNR	$K = 0.75, \sigma = 5$ SSIM
LSD2 [11]	dual-exposure Quad-Bayer	113.433G	0.0079	32.21	0.9264	32.67	0.9317	32.81	0.9333
MWCNN [58]	dual-exposure Quad-Bayer	301.671G	0.1514	31.85	0.9193	32.28	0.9244	32.47	0.9278
SGN [91]	dual-exposure Quad-Bayer	51.258G	0.0113	32.00	0.9227	32.46	0.9281	32.64	0.9299
MIRNet [62]	dual-exposure Quad-Bayer	2.607T	0.6869	32.18	0.9275	32.72	0.9326	32.87	0.9339
MPRNet [31]	dual-exposure Quad-Bayer	2.642T	0.1435	32.24	0.9267	32.41	0.9301	32.81	0.9323
SRN [29]	dual-exposure Quad-Bayer	343.004G	0.0545	32.42	0.9296	32.86	0.9345	33.01	0.9358
DeblurGAN [8]	dual-exposure Quad-Bayer	205.445G	0.0111	32.55	0.9303	32.93	0.9336	33.15	0.9360
DMPHN [30]	dual-exposure Quad-Bayer	202.458G	0.0995	32.51	0.9303	32.93	0.9344	33.10	0.9359
MIMOUNet [10]	dual-exposure Quad-Bayer	300.205G	0.0394	32.71	0.9332	33.09	0.9366	33.31	0.9387
MIMOUNet++ [10]	dual-exposure Quad-Bayer	687.105G	0.0831	32.82	0.9350	33.26	0.9386	33.40	0.9398
QRNet	dual-exposure Quad-Bayer	34.628G	0.0178	33.24	0.9403	33.61	0.9428	33.81	0.9445

the edge differences. In detail, we first apply white balance. The gain for green channel  $w_g$  is fixed to 1. The gains for red channel  $w_r$  and blue channel  $w_b$  are sampled uniformly from [1.6, 1.8] and [1.4, 1.6], respectively. Then, we apply gamma correction with  $\lambda = 2.2$ , which is an exponential calculation. Next, we demosaic them with [92]. After that, we apply the MATLAB “imsharpen” tool to post-process them. Finally, we apply the same sampling scheme and obtain  $y$ . We also illustrate the transformed RGB images with the same demosaicking and sampling schemes corresponding to  $x_l$  and  $4x_s$  (i.e., the intensity of  $4x_s$  is four times larger than  $x_s$ ) in Figure 5 (b), respectively.

The overall synthesis workflow results in 731 tuples  $(x_s, x_l, x, y_s, y_l, y)$ , where every dual-exposure Quad-Bayer image ( $x$ ) and sharpened RGB image ( $y$ ) forms a pair. They have a resolution of  $3016 \times 2008$ . They are divided into 701 training tuples and 30 validation tuples.

In addition, we capture 24 real Quad-Bayer images with the Sony IMX 586 Quad-Bayer module for testing. The data arrangement is identical to our synthetic data. The captured images are all indoor scenes. The image resolution is  $4608 \times 3456$  and the data is in 10 bits.

## V. EXPERIMENT

### A. Implementation Details

We train the QRNet with 200 epochs and each epoch has 2500 iterations. The batch size is 16. To facilitate the training, we select four full-resolution images first and then crop four  $320 \times 320$  patches from each full-resolution image. The learning rate is initialized as  $1 \times 10^{-4}$ . During the 100-th epoch to the 200-th epoch, the learning rate is linearly decreased to  $5 \times 10^{-7}$ . The frequency loss parameter  $\alpha$  is set to 0.01. We use Adam optimizer [78] with  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$ . All trainable parameters of QRNet are initialized to random values from the standard normal distribution. We implement the QRNet with PyTorch 1.0.0 and CUDA 9.0. We train it on 4 NVIDIA TITAN Xp GPUs. It takes approximately 30 hours to complete the whole training process.

### B. Experiment Details

**Experiment settings.** To perform a comprehensive and fair comparison, we first compare the proposed QRNet with other methods from degraded dual-exposure Quad-Bayer to clean and sharp RGB  $y$ . Then, we compare them on four more pipelines:

- 1) Short-exposure-RAW-based methods: From degraded short-exposure Quad-Bayer to clean and sharp RGB  $y$ ;
- 2) Long-exposure-RAW-based methods: From degraded long-exposure Quad-Bayer to clean and sharp RGB  $y$ ;
- 3) Short-exposure-RGB-based methods: From degraded short-exposure RGB to clean and sharp RGB  $y$ ;
- 4) Long-exposure-RGB-based methods: From degraded long-exposure RGB to clean and sharp RGB  $y$ .

For the training of other methods, the data selection scheme, training epochs, and batch size are identical to QRNet for fair comparisons. Since the synthetic tuples  $x_s, x_l, x, y_s, y_l, y$  are almost noise-free, we add noises with different levels to the synthetic data for benchmarking, which matches the real-world application scenarios.

**Noise modeling.** We add the noise  $n$  in the RAW space based on the physical model [4], which can be represented as:

$$\begin{aligned} n &= n_s + n_r + n_q, \\ x_* + n_s &\sim \mathcal{P}(x_* \cdot K)/K, \\ n_r &\sim \mathcal{G}(0, \sigma^2), \\ n_q &\sim \mathcal{U}(-1/2qu, 1/2qu), \end{aligned} \quad (6)$$

where  $x_*$  is a clean RAW image.  $n_s$ ,  $n_r$ , and  $n_q$  are shot, read, and quantization noises, respectively.  $\mathcal{P}$ ,  $\mathcal{G}$ , and  $\mathcal{U}$  denote Poisson, Gaussian, and Uniform distributions, respectively. Here  $K$  and  $\sigma$  are noise model hyper-parameters, where we set  $K = 0.25$  or  $0.5$  or  $0.75$  and  $\sigma = 5$  in the experiment. They simulate different noise levels.  $qu$  is a static quantization step. After adding noises to the RAW image, we multiply the exposure ratio  $A$  with short-exposure pixels.

Note if inputs are RGB images ( $y_s$  or  $y_l$ ), we first convert them to RAW space to add noises and then transform them

TABLE III

THE COMPARISONS OF QRNET AND MIMOUNET++ [10] BASED ON DUAL-EXPOSURE QUAD-BAYER IMAGES. MORE NOISE PARAMETER SETTINGS ARE INCLUDED. THE RED COLOR HIGHLIGHTS THE BETTER RESULTS AMONG QRNET AND MIMOUNET++.

Method	$K = 0.75, \sigma = 5$	$K = 5, \sigma = 5$	$K = 0.75, \sigma = 3$	$K = 5, \sigma = 3$	$K = 0.75, \sigma = 1$	$K = 5, \sigma = 1$
	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
MIMOUNet++ [10]	33.40 0.9398	33.84 0.9439	34.53 0.9486	35.29 0.9536	35.85 0.9578	37.38 0.9653
QRNet	<b>33.81</b> <b>0.9445</b>	<b>34.19</b> <b>0.9471</b>	<b>34.91</b> <b>0.9523</b>	<b>35.60</b> <b>0.9563</b>	<b>36.27</b> <b>0.9617</b>	<b>37.65</b> <b>0.9672</b>

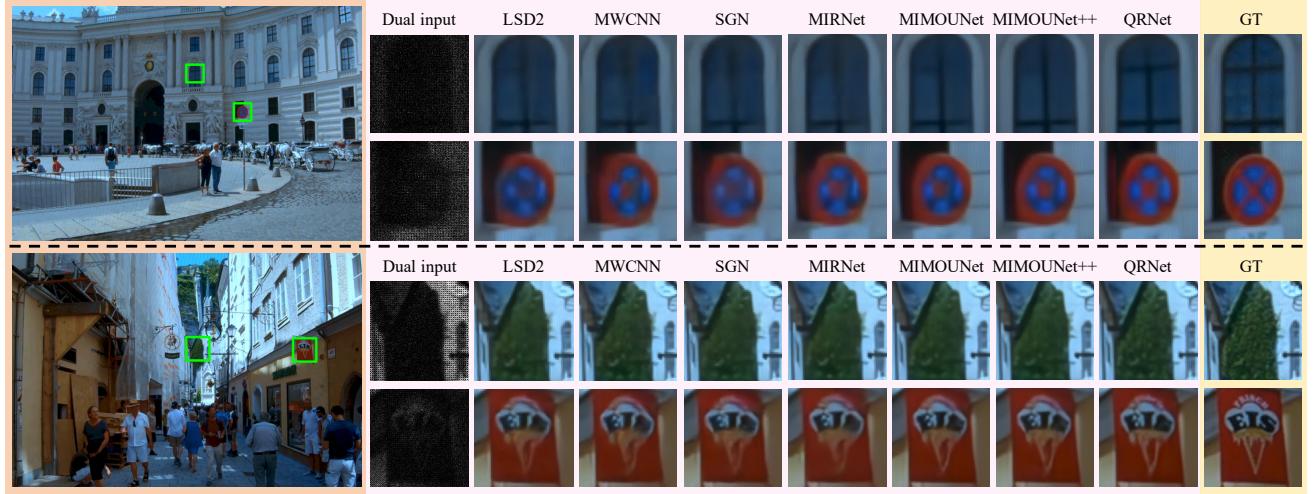


Fig. 6. Illustration of restoration results of QRNet and other methods based on dual-exposure Quad-Bayer images. The noise parameter is  $K = 0.25, \sigma = 5$ . The left is full-resolution ground truth images. On the right, input dual-exposure Quad-Bayer image patches and generated image patches are in the pink background. Ground truth image patches are in the yellow background.

back to RGB images [1] at training.

$$\begin{aligned}\hat{y}_s &= RAW2RGB(A \cdot (RGB2RAW(y_s)/A + n)), \\ \hat{y}_l &= RAW2RGB(RGB2RAW(y_l) + n),\end{aligned}\quad (7)$$

where  $\hat{y}_s$  and  $\hat{y}_l$  are noisy versions of  $y_s$  and  $y_l$  for training.  $RAW2RGB$  contains demosaicking, white balance, and gamma correction and  $RGB2RAW$  contains the inverse transformations.  $n$  is the added noises as shown in Equation 6.

**Evaluation metrics.** We employ PSNR and SSIM metrics [94] to assess the pairwise image quality across 30 validation tuples within the QR dataset. Additionally, we determine the Multiply-Accumulate (MACs) using the PyTorch-OpCounter [95] library. The processing time for a  $512 \times 512$  patch on an NVIDIA RTX 3090 GPU is calculated by averaging the time taken to forward 10000 patches for ten repetitions. These computations were performed using PyTorch 1.7.0 and CUDA 11.0.

### C. Evaluation with Dual-exposure-based Methods

In this section, we evaluate the image restoration quality of QRNet and state-of-the-art methods including LSD2 [11], MWCNN [58], SGN [91], MIRNet [62], SRN [29], DeblurGAN [8], DMPHN [30], MIMOUNet, and MIMOUNet++ [10] based on dual-exposure Quad-Bayer data.

First, we conduct the quantitative analysis on the QR validation set, where the results are concluded in Table II. It is clear that QRNet has the least MACs, and the highest PSNR and SSIM across all the methods. QRNet also outperforms all other methods under different noise levels, which demonstrates

its stability. Compared with the state-of-the-art method MIMOUNet++, QRNet only has 5% MACs of it and obtains 0.32 higher PSNR under  $K = 0.25, \sigma = 5$  setting. Compared with the other fast method SGN, QRNet obtains 1.14 higher PSNR and 0.0158 higher SSIM under  $K = 0.25, \sigma = 5$  setting. To further demonstrate the superior image restoration quality of QRNet, we conduct 5 more experiments as concluded in Table III. QRNet obtains better PSNR and SSIM values than the state-of-the-art method MIMOUNet++ even under more different noise levels. The experiments demonstrate the QRNet architecture is advanced for restoring Quad-Bayer sensor data.

We show some generated images in Figure 6. The results from QRNet are sharper, e.g., the crossbar of the window is recovered by QRNet while other methods do not in the first row; there are fewer color bleeding artifacts for QRNet in the second row; the textures and edges are more obvious for QRNet in the third and fourth rows. They show the QRNet has a better ability to restore details and textures than existing methods under the same noise parameter in the training.

Secondly, we evaluate the QRNet and other methods [8], [10], [30] on real Quad-Bayer images captured by the IMX586 module. The noise parameter is  $K = 5, \sigma = 1$  for all the methods at training, which is estimated on real Quad-Bayer images based on [4]. We show two blurry samples (top two rows) and two highly noisy samples (bottom two rows) in Figure 7. On one hand, since there is no ground truth for real Quad-Bayer images, we adopt the non-reference image quality evaluator LIQE [93] to measure the quality of all generated images, which better aligns with mean opinion scores (MOSs)

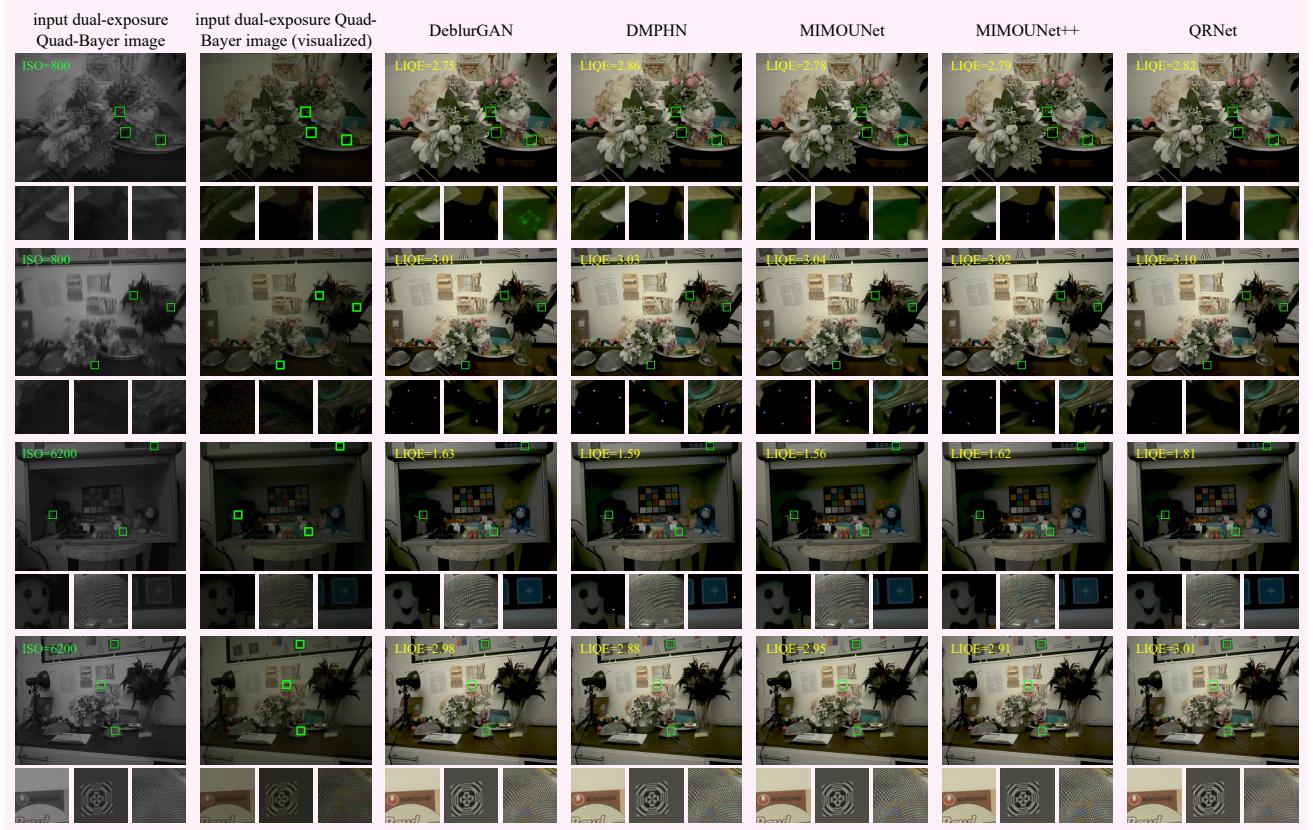


Fig. 7. Illustration of 4 real input Quad-Bayer images and restoration results from different methods. For every full-resolution image, three patches are extracted and named  $\#1$ ,  $\#2$ , and  $\#3$ . The left column includes input Quad-Bayer images, where ISO values are shown at the top left corner. The second-left column includes the visualization results by demosaicking the Bayer counterparts of input images. The other columns include restoration results from DeblurGAN [8], DPMHN [30], MIMOUNet [10], MIOMUNet++ [10], and QRNet, where LIQE [93] values are shown at the top left corner.

than previous evaluators. The LIQE values are illustrated on the top left corner of every generated image in Figure 7. For all 4 samples, QRNet has higher LIQE values than other methods, which denotes that QRNet’s results better match perceptual quality. On the other hand, we notice that other methods produce more abnormal pixels than QRNet, e.g., patches  $\#1$ - $\#2$  of row 1, patches  $\#1$ - $\#3$  of row 2, and patch  $\#3$  of row 3. Compared with other methods, QRNet produces less color artifacts, e.g., patch  $\#2$  of row 3 and patch  $\#3$  of row 4. In addition, QRNet generates clearer words, e.g., “NONGSHIM” in patch  $\#1$  of row 4.

From the experiment results, we observe that QRNet outperforms existing methods on both synthetic and real dual-exposure Quad-Bayer images. We assume the QRNet architecture (including the input enhancement block, multi-level feature extraction, and inter-level feature interactions) can better restore RAW images with the Quad-Bayer pattern compared with existing methods.

#### D. Evaluation on Quad-Bayer RAW Image Restoration

In this section, we conduct experiments on joint deblurring and denoising in the Quad-Bayer RAW domain. The input is a blurry and noisy Quad-Bayer RAW image while the output is a sharp and clean one. Compared with RAW to RGB mapping, restoring RAW images enables the adjustment of the

TABLE IV  
THE COMPARISONS OF QRNET AND OTHER METHODS ON DUAL-EXPOSURE QUAD-BAYER RAW IMAGE RESTORATION.

Method	$K = 0.25, \sigma = 5$		$K = 0.5, \sigma = 5$		$K = 0.75, \sigma = 5$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LSD2 [11]	39.20	0.9907	40.26	0.9918	40.75	0.9923
MWCNN [58]	39.14	0.9907	40.23	0.9919	40.73	0.9923
SGN [91]	38.72	0.9899	40.01	0.9915	40.49	0.9919
MIRNet [62]	39.03	0.9905	40.15	0.9917	40.64	0.9922
MPRNet [31]	39.09	0.9906	40.42	0.9922	40.92	0.9926
SRN [29]	39.63	0.9916	40.47	0.9923	41.19	0.9930
DeblurGAN [8]	39.07	0.9906	40.05	0.9917	40.58	0.9922
DMPHN [30]	39.17	0.9908	39.98	0.9916	40.55	0.9921
MIMOUNet [10]	<b>39.64</b>	<b>0.9916</b>	<b>40.70</b>	<b>0.9926</b>	<b>41.22</b>	<b>0.9931</b>
MIMOUNet++ [10]	<b>39.62</b>	<b>0.9915</b>	<b>40.72</b>	<b>0.9926</b>	<b>41.27</b>	<b>0.9931</b>
QRNet	<b>39.88</b>	<b>0.9919</b>	<b>40.88</b>	<b>0.9928</b>	<b>41.35</b>	<b>0.9932</b>

following ISP steps such as white balance, tone mapping, color adjustment, etc. Specifically, we compare QRNet with methods described in Section V-C while the target changes to clean and sharp Quad-Bayer data. The quantitative results are concluded in Table IV, where QRNet still achieves higher PSNR and SSIM values than other methods under 3 different noise levels. It further demonstrates that QRNet architecture can better restore Quad-Bayer images even having less computation costs.

TABLE V  
THE COMPARISONS OF QRNET AND OTHER BASELINES BASED ON SHORT-EXPOSURE IMAGES.

Method	Input data structure	MACs (512×512)	Running time (sec / 512×512)	$K = 0.25, \sigma = 5$		$K = 0.5, \sigma = 5$		$K = 0.75, \sigma = 5$	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
IRCNN [54]	short-exposure Quad-Bayer	54.459G	0.0026	28.99	0.8914	29.23	0.8930	29.42	0.8955
DnCNN [55]	short-exposure Quad-Bayer	162.001G	0.0057	28.82	0.8917	29.26	0.8939	29.12	0.8936
MemNet [56]	short-exposure Quad-Bayer	194.986G	0.0305	15.46	0.7517	15.26	0.7490	15.62	0.7588
MWCNN [58]	short-exposure Quad-Bayer	301.671G	0.1514	31.71	0.9177	31.89	0.9187	31.98	0.9197
SGN [91]	short-exposure Quad-Bayer	51.258G	0.0113	31.78	0.9183	32.16	0.9231	32.27	0.9240
DSWN [61]	short-exposure Quad-Bayer	102.283G	0.0116	31.24	0.9114	31.48	0.9141	31.57	0.9134
ADNet [60]	short-exposure Quad-Bayer	137.158G	0.0066	16.71	0.7870	16.98	0.7905	17.10	0.7954
RIDNet [59]	short-exposure Quad-Bayer	439.379G	0.0182	30.09	0.9062	30.34	0.9088	30.29	0.9084
MIRNet [62]	short-exposure Quad-Bayer	2.607T	0.6869	31.36	0.9079	32.17	0.9237	32.34	0.9261
MPRNet [31]	short-exposure Quad-Bayer	2.642T	0.1435	30.19	0.9089	30.13	0.9060	29.65	0.8969
HINet [96]	short-exposure Quad-Bayer	769.487G	0.0315	24.15	0.8315	27.62	0.8810	21.92	0.8054
QRNet	short-exposure Quad-Bayer	34.628G	0.0178	32.89	0.9372	33.19	0.9400	33.33	0.9411
IRCNN [54]	short-exposure RGB	54.794G	0.0026	29.86	0.8927	30.03	0.8942	30.16	0.8945
DnCNN [55]	short-exposure RGB	162.336G	0.0059	29.89	0.8939	30.04	0.8946	30.21	0.8955
MemNet [56]	short-exposure RGB	195.140G	0.0309	16.59	0.7723	16.86	0.7829	16.73	0.7850
MWCNN [58]	short-exposure RGB	301.671G	0.1517	31.20	0.9134	31.45	0.9160	31.54	0.9172
SGN [91]	short-exposure RGB	53.774G	0.0116	31.48	0.9157	31.81	0.9205	31.96	0.9221
DSWN [61]	short-exposure RGB	104.800G	0.0113	30.71	0.9063	31.19	0.9105	31.21	0.9116
ADNet [60]	short-exposure RGB	137.460G	0.0065	18.06	0.7743	17.10	0.7835	18.51	0.7977
RIDNet [59]	short-exposure RGB	439.714G	0.0198	30.80	0.9095	31.03	0.9098	31.11	0.9091
MIRNet [62]	short-exposure RGB	2.607T	0.6893	31.67	0.9193	31.89	0.9219	32.03	0.9235
MPRNet [31]	short-exposure RGB	2.642T	0.1516	30.44	0.9043	31.24	0.9127	31.32	0.9136
HINet [96]	short-exposure RGB	769.487G	0.0313	30.32	0.8903	27.24	0.8253	28.78	0.8791
QRNet	dual-exposure Quad-Bayer	34.628G	0.0178	33.24	0.9403	33.61	0.9428	33.81	0.9445

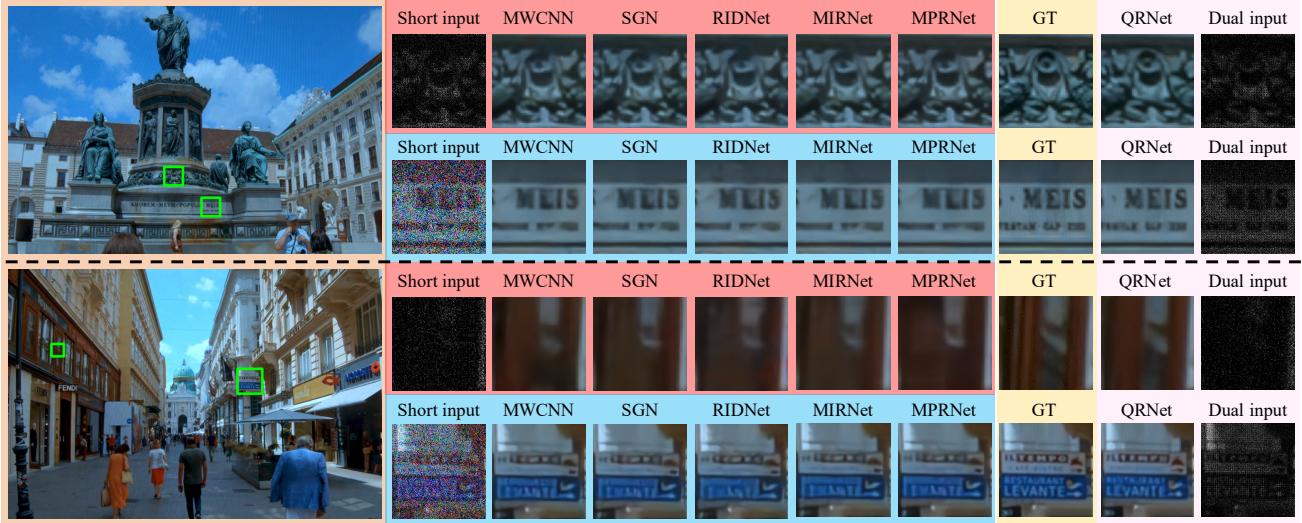


Fig. 8. Illustration of image restoration results of QRNet and other methods based on short-exposure images. The noise parameter is  $K = 0.25, \sigma = 5$ . The left is full-resolution ground truth images. On the right, input short-exposure Quad-Bayer image patches and generated image patches are in the red background; corresponding input short-exposure RGB image patches and generated image patches are in the blue background; corresponding input dual-exposure Quad-Bayer image patches and generated image patches are in the pink background. Ground truth image patches are in the yellow background.

#### E. Evaluation with Short-exposure-based Methods

In this section, we evaluate the QRNet based on the pipeline from dual-exposure Quad-Bayer images to clean and sharp RGB  $y$  with methods based on short-exposure Quad-Bayer images or RGB images. The methods include IRCNN [54], DnCNN [55], MemNet [56], MWCNN [58], SGN [91], DSWN [61], ADNet [60], RIDNet [59], MIRNet [62], MPRNet [31], and HINet [96].

The quantitative analysis is concluded in Table V. From the results, we draw four findings. First, the dual-exposure Quad-Bayer is superior for image denoising than only short

exposure since it also contains long-exposure pixels, which are less noisy. Neural networks can utilize such information to improve the details and textures. For instance, SGN obtains a PSNR of 32.00 for dual exposure, while it is 31.78 (-0.22) for short exposure when  $K = 0.25, \sigma = 5$ . Secondly, Quad-Bayer is better for image restoration than RGB images since RAW data contains more information than the RGB counterpart. The top-performing method MIRNet [62] based on short-exposure RGB obtains PSNR of 31.67, while SGN [91] based on short-exposure Quad-Bayer obtains PSNR of 32.89 (+1.22), although it uses much fewer MACs than MIRNet. For SGN, it obtains PSNR of 31.78 when receiving Quad-

TABLE VI  
THE COMPARISONS OF QRNET AND OTHER BASELINES BASED ON LONG-EXPOSURE IMAGES.

Method	Input data structure	MACs (512×512)	Running time (sec / 512×512)	$K = 0.25, \sigma = 5$		$K = 0.5, \sigma = 5$		$K = 0.75, \sigma = 5$	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DeepDeblur [28]	long-exposure Quad-Bayer	1.398T	0.0366	26.37	0.8685	26.66	0.8757	25.30	0.8243
SRN [29]	long-exposure Quad-Bayer	343.004G	0.0545	27.56	0.8988	27.66	0.9007	27.72	0.9027
DeblurGAN [8]	long-exposure Quad-Bayer	205.445G	0.0111	27.40	0.8967	27.51	0.8990	27.55	0.9001
DeblurGANv2 [9]	long-exposure Quad-Bayer	52.730G	0.0249	21.66	0.8439	21.08	0.8438	21.45	0.8474
DMPHN [30]	long-exposure Quad-Bayer	202.458G	0.0995	27.50	0.8982	27.67	0.9010	27.68	0.9019
MPRNet [31]	long-exposure Quad-Bayer	2.642T	0.1435	27.07	0.8915	27.20	0.8954	27.19	0.8962
HINet [96]	long-exposure Quad-Bayer	769.487G	0.0315	27.08	0.8880	25.89	0.8143	25.08	0.8022
MIMOUNet [10]	long-exposure Quad-Bayer	300.205G	0.0394	27.60	0.9007	27.73	0.9030	27.79	0.9041
MIMOUNet++ [10]	long-exposure Quad-Bayer	687.105G	0.0831	27.64	0.9014	27.76	0.9034	27.81	0.9048
QRNet	long-exposure Quad-Bayer	34.628G	0.0178	27.78	0.9031	27.95	0.9058	27.94	0.9062
DeepDeblur [28]	long-exposure RGB	1.398T	0.0365	27.32	0.8932	27.06	0.8857	26.93	0.8827
SRN [29]	long-exposure RGB	343.004G	0.0563	27.68	0.9006	27.82	0.9029	27.86	0.9037
DeblurGAN [8]	long-exposure RGB	207.123G	0.0092	27.39	0.8966	27.44	0.8982	27.54	0.8996
DeblurGANv2 [9]	long-exposure RGB	54.408G	0.0258	20.51	0.8308	20.64	0.8350	20.51	0.8368
DMPHN [30]	long-exposure RGB	202.458G	0.0998	17.04	0.7968	16.06	0.7604	17.88	0.8006
MPRNet [31]	long-exposure RGB	2.642T	0.1516	27.34	0.8928	27.49	0.8959	27.57	0.8981
HINet [96]	long-exposure RGB	769.487G	0.0313	21.53	0.8132	23.60	0.8396	22.20	0.8169
MIMOUNet [10]	long-exposure RGB	300.205G	0.0397	27.60	0.9008	27.67	0.9024	27.78	0.9037
MIMOUNet++ [10]	long-exposure RGB	687.105G	0.0835	27.64	0.9013	27.71	0.9025	27.74	0.9032
QRNet	dual-exposure Quad-Bayer	34.628G	0.0178	33.24	0.9403	33.61	0.9428	33.81	0.9445

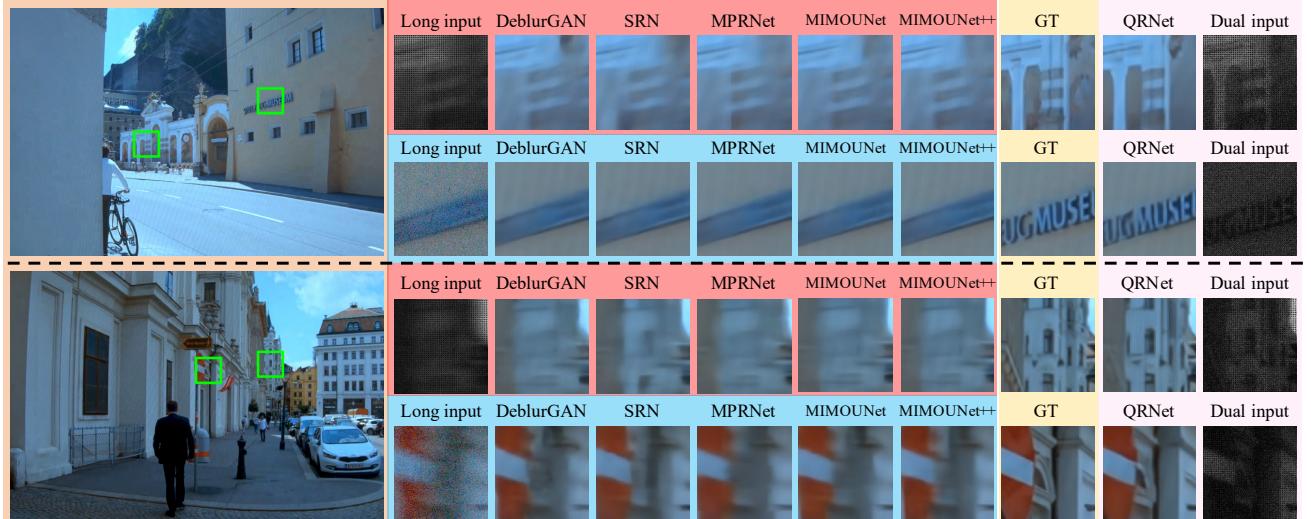


Fig. 9. Illustration of image restoration results of QRNet and other baselines based on long-exposure images. The noise parameter is  $K = 0.75, \sigma = 5$ . The left is full-resolution ground truth images. On the right, input long-exposure Quad-Bayer image patches and generated image patches are in the red background; corresponding input long-exposure RGB image patches and generated image patches are in the blue background; corresponding input dual-exposure Quad-Bayer image patches and generated image patches are in the pink background. Ground truth image patches are in the yellow background.

Bayer while obtains 31.48 when receiving RGB images when  $K = 0.25, \sigma = 5$ . Thirdly, QRNet achieves better performance than existing methods by a large margin under different noise levels. It reveals that QRNet has good generalization ability. Finally, when changing the input data structure to the short-exposure Quad-Bayer for the QRNet, it still outperforms other methods. It further demonstrates that the improvement is induced by the QRNet network architecture.

We illustrate some restoration results in Figure 8. The results of QRNet are clearer and sharper than short-exposure-based methods. For instance, the letters “MEIS” in the second row and the letters “LEVANTE” in the fourth row of QRNet are much clearer than MWCNN, SGN, and MPRNet. This is because QRNet can use the information from long-exposure pixels, which are less noisy than short-exposure pixels. How-

ever, short-exposure-based methods can only exploit information from short-exposure pixels either in the Quad-Bayer RAW domain or RGB domain. Therefore, their performance is restricted when recovering the details and textures, while QRNet with dual-exposure Quad-Bayer can well address these problems.

#### F. Evaluation with Long-exposure-based Methods

In this section, we evaluate the QRNet based on the pipeline from dual-exposure Quad-Bayer images to clean and sharp RGB  $y$  with methods based on long-exposure Quad-Bayer images or RGB images. The methods include DeepDeblur [28], SRN [29], DeblurGAN [8], DeblurGANv2 [9], DPMHN [30], MPRNet [31], HINet [96], MIMOUNet [10], and MIMOUNet++ [10].

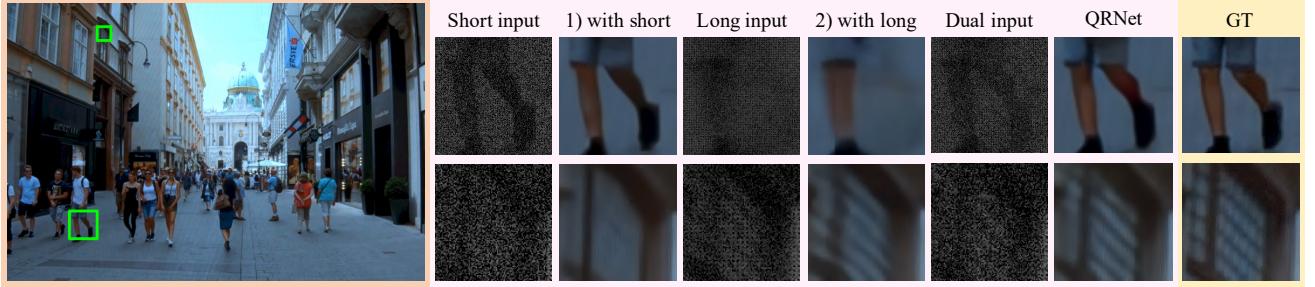


Fig. 10. Illustration of image restoration results of QRNet and changing input Quad-Bayer data structure ablation study settings. The noise parameter is  $K = 0.25, \sigma = 5$ . The left is full-resolution ground truth images. On the right, three pairs of input Quad-Bayer image patches and generated image patches (including ablation study settings and full QRNet) are in the pink background. Ground truth image patches are in the yellow background.

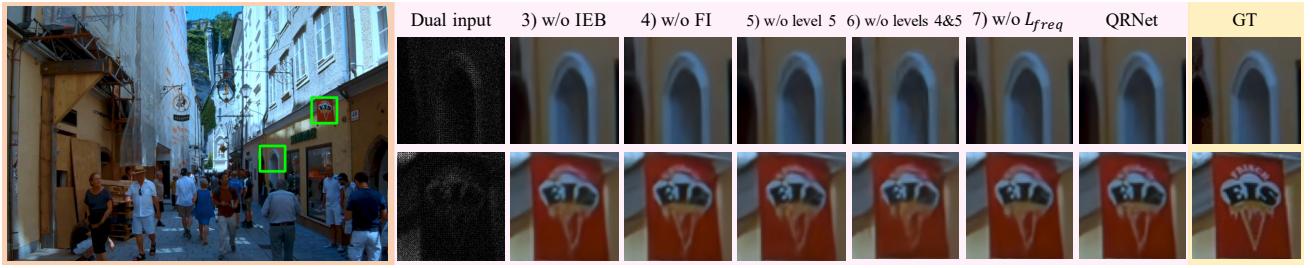


Fig. 11. Illustration of image restoration results of QRNet and other ablation study settings. The noise parameter is  $K = 0.25, \sigma = 5$ . The left is full-resolution ground truth images. In the right, input dual-exposure Quad-Bayer image patches and generated image patches (including ablation study settings and full QRNet) are in the pink background. Ground truth image patches are in the yellow background.

The quantitative analysis is concluded in Table VI. On one hand, the proposed QRNet performs better than other long-exposure-based methods. Since the short-exposure pixels are captured in the dual-exposure Quad-Bayer, they can help deblurring due to no motion blur, although they include more noises. QRNet can well utilize this information to estimate the motion blur. Moreover, QRNet still outperforms other methods when receiving long-exposure Quad-Bayer input, e.g., QRNet obtains PSNR of 27.78 while MIMOUNet++ obtains 27.64 when  $K = 0.25, \sigma = 5$ . On the other hand, the state-of-the-art network architecture MIMOUNet++ trained on dual-exposure Quad-Bayer is still inferior to QRNet, even under different noise parameters, as concluded in Table III. It demonstrates that the QRNet architecture is more appropriate for restoring dual-exposure Quad-Bayer data.

The qualitative analysis is illustrated in Figure 9. When there is motion blur in the captured images, which is common in real-world scenarios, QRNet is robust enough to recover sharp and clean images. However, the image deblurring methods based on either single-exposure Quad-Bayer RAW or RGB images cannot handle large blurs.

### G. Ablation Study

To demonstrate the effectiveness of several components of QRNet, we define some ablation study settings:

- 1) with short: replacing dual-exposure Quad-Bayer input with short-exposure Quad-Bayer input;
- 2) with long: replacing dual-exposure Quad-Bayer input with long-exposure Quad-Bayer input;

TABLE VII  
THE ABLATION STUDY COMPARISONS OF QRNET.

Setting	Description	MACs (512×512)	$K = 0.25, \sigma = 5$
		PSNR	SSIM
with short	Input data	34.628G	32.89 0.9372
with long	Input data	34.628G	27.78 0.9031
w/o IEB	Network	33.084G	33.15 0.9391
w/o FI	Network	31.944G	33.14 0.9391
w/o level 5	Network	21.539G	32.79 0.9349
w/o level 4&5	Network	12.475G	31.97 0.9238
w/o $L_{freq}$	Loss function	34.628G	33.13 0.9389
QRNet	Full QRNet	34.628G	<b>33.24</b> <b>0.9403</b>

- 3) w/o IEB: replacing input enhancement block (IEB) with a convolutional layer receiving only Pixel Unshuffle downsampled data;
- 4) w/o FI: dropping all the feature interactions of levels 3 to 5;
- 5) w/o level 5: dropping all layers of level 5;
- 6) w/o level 4&5: dropping all layers of levels 4 and 5;
- 7) w/o  $L_{freq}$ : dropping the frequency loss  $L_{freq}$  ( $\alpha = 0$ ).

We conduct the experiments on the QR dataset validation set. The quantitative results are concluded in Table VII and some results are illustrated in Figure 10 and 11.

**Input data.** Dual-exposure Quad-Bayer data is the most significant for QRNet to obtain the performance gain. Compared with only using short- or long-exposure data, dual exposure brings 0.35dB and 5.46dB PSNR gains. It shows that dual-exposure Quad-Bayer is superior to single exposure since multiple exposures can provide more information. As shown in Figure 10, QRNet with dual-exposure Quad-Bayer produces

sharper outputs than setting 1), e.g., the window. As the long-exposure pixels are less noisy, QRNet can extract the texture and detail information from them. The QRNet outputs are also less blurry than setting 2), e.g., the leg. Since the short-exposure pixels provide accurate position information, QRNet can better estimate the motion field with dual-exposure data than only using long-exposure pixels.

**Network.** All the proposed input enhancement block (IEB), feature interaction (FI), and multi-level feature extraction are helpful in improving image restoration quality. For instance, settings 5) and 6) generate a deformed archdoor (first row); and settings 3-6) produce blurry flags (second row), as shown in Figure 11. However, the results full QRNet have more distinguishable and sharper edges. From Table VII, we can see IEB, FI, level 5 of QRNet, and level 4&5 of QRNet bring 0.09dB, 0.1dB, 0.45dB, and 1.27dB PSNR gains. They demonstrate every network component is helpful in recovering textures and details.

**Loss function.** From setting 7), we can see dropping  $L_{freq}$  leads to decreases of PSNR (-0.11dB) and SSIM metrics. Also as shown in Figure 11, the edges at the top of the flag of setting 7) are blurrier than the full QRNet. It demonstrates that frequency loss can enhance detail recovery.

In conclusion, the dual-exposure data structure, network components, and loss function are beneficial to improving the image restoration quality for QRNet.

## VI. CONCLUSION

In this paper, we have presented a novel pipeline for joint image deblurring and denoising of Quad-Bayer RAW images. The Quad-Bayer pattern provides complementary exposures that enable both noise reduction and motion deblurring. To effectively restore clean and sharp RGB images from Quad-Bayer inputs, we have proposed QRNet, an efficient hierarchical neural network architecture. A key component of QRNet is the input enhancement block which reduces artifacts from downsampling and enhances details. Inter-level feature interactions further improve feature representations by sharing information between network levels. We have also introduced a new Quad-Bayer to RGB mapping dataset (QR dataset) to train and benchmark QRNet. Experiments demonstrate state-of-the-art performance in joint deblurring and denoising using our approach. The proposed QRNet achieves superior image restoration quality compared to existing methods while using less computational costs.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and editors for many helpful comments. The authors would also like to thank the computational imaging group of Sense-Time Research and Tetras.AI for hardware support and helpful discussions.

## REFERENCES

- [1] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, “Unprocessing images for learned raw denoising,” in *Proc. CVPR*, 2019, pp. 11 036–11 045.
- [2] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, “Cycleisp: Real image restoration via improved data synthesis,” in *Proc. CVPR*, 2020, pp. 2696–2705.
- [3] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, “Noise flow: Noise modeling with conditional normalizing flows,” in *Proc. ICCV*, 2019, pp. 3165–3173.
- [4] K. Wei, Y. Fu, J. Yang, and H. Huang, “A physics-based noise formation model for extreme low-light raw denoising,” in *Proc. CVPR*, 2020, pp. 2758–2767.
- [5] Y. Wang, H. Huang, Q. Xu, J. Liu, Y. Liu, and J. Wang, “Practical deep raw image denoising on mobile devices,” in *Proc. ECCV*, 2020, pp. 1–16.
- [6] Y. Zhang, H. Qin, X. Wang, and H. Li, “Rethinking noise synthesis and modeling in raw denoising,” in *Proc. ICCV*, 2021, pp. 4593–4601.
- [7] A. Maleky, S. Kousha, M. S. Brown, and M. A. Brubaker, “Noise2noiseflow: Realistic camera noise modeling without clean images,” in *Proc. CVPR*, 2022, pp. 17 611–17 620.
- [8] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, “Deblurgan: Blind motion deblurring using conditional adversarial networks,” in *Proc. CVPR*, 2018, pp. 8183–8192.
- [9] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, “Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better,” in *Proc. ICCV*, 2019, pp. 8878–8887.
- [10] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, “Rethinking coarse-to-fine approach in single image deblurring,” in *Proc. ICCV*, 2021, pp. 4641–4650.
- [11] J. Mustanemi, J. Kannala, J. Matas, S. Särkkä, and J. Heikkilä, “Lsd2 - joint denoising and deblurring of short and long exposure images with convolutional neural networks,” in *Proc. BMVC*, 2020.
- [12] M. Chang, H. Feng, Z. Xu, and Q. Li, “Low-light image restoration with short-and long-exposure raw pairs,” *IEEE Trans. Multimedia*, vol. 24, pp. 702–714, 2021.
- [13] Y. Zhao, Y. Xu, Q. Yan, D. Yang, X. Wang, and L.-M. Po, “D2hnet: Joint denoising and deblurring with hierarchical network for robust night image restoration,” in *Proc. ECCV*, 2022, pp. 91–110.
- [14] I. Kim, Y. Choi, H. Ko, D. Lim, Y. Seo, J. Lee, G. Lee, E. Heo, S. Song, and S. Lim, “Under display camera quad bayer raw image restoration using deep learning,” *Electronic Imaging*, vol. 2021, no. 7, pp. 67–1, 2021.
- [15] Y. Jiang, I. Choi, J. Jiang, and J. Gu, “Hdr video reconstruction with tri-exposure quad-bayer sensors,” *arXiv preprint arXiv:2103.10982*, 2021.
- [16] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proc. CVPR*, 2016, pp. 1874–1883.
- [17] B. E. Bayer, “Color imaging array,” U.S. Patent 3971065, Jul. 1976.
- [18] R. Lukac and K. N. Plataniotis, “Color filter arrays: Design and performance analysis,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1260–1267, 2005.
- [19] S. Yamanaka, “Solid state camera,” U.S. Patent 4054906, Nov. 1977.
- [20] M. Parmar and S. J. Reeves, “A perceptually based design methodology for color filter arrays [image reconstruction],” in *Proc. ICASSP*, vol. 3, 2004, pp. iii–473.
- [21] W. H. Richardson, “Bayesian-based iterative method of image restoration,” *JoSA*, vol. 62, no. 1, pp. 55–59, 1972.
- [22] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” in *Proc. ACM SIGGRAPH*, 2006, pp. 787–794.
- [23] L. Sun, S. Cho, J. Wang, and J. Hays, “Edge-based blur kernel estimation using patch priors,” in *Proc. ICCP*, 2013, pp. 1–8.
- [24] D. Krishnan, T. Tay, and R. Fergus, “Blind deconvolution using a normalized sparsity measure,” in *Proc. CVPR*, 2011, pp. 233–240.
- [25] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, “Deblurring text images via 10-regularized intensity and gradient prior,” in *Proc. CVPR*, 2014, pp. 2901–2908.
- [26] J. Sun, W. Cao, Z. Xu, and J. Ponce, “Learning a convolutional neural network for non-uniform motion blur removal,” in *Proc. CVPR*, 2015, pp. 769–777.
- [27] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, “From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur,” in *Proc. CVPR*, 2017, pp. 2319–2328.
- [28] S. Nah, T. Hyun Kim, and K. Mu Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in *Proc. CVPR*, 2017, pp. 3883–3891.
- [29] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, “Scale-recurrent network for deep image deblurring,” in *Proc. CVPR*, 2018, pp. 8174–8182.

- [30] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. CVPR*, 2019, pp. 5978–5986.
- [31] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. CVPR*, 2021, pp. 14821–14831.
- [32] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M.-H. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *Proc. CVPR*, 2018, pp. 2521–2529.
- [33] Y. Yuan, W. Su, and D. Ma, "Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training," in *Proc. CVPR*, 2020, pp. 3555–3564.
- [34] D. Park, D. U. Kang, J. Kim, and S. Y. Chun, "Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training," in *Proc. ECCV*, 2020, pp. 327–343.
- [35] Z. Chen, Y. Zhang, D. Liu, B. Xia, J. Gu, L. Kong, and X. Yuan, "Hierarchical integration diffusion model for realistic image deblurring," in *Proc. NeurIPS*, 2022, pp. 29114–29125.
- [36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.
- [37] H. Chen, J. Gu, O. Gallo, M.-Y. Liu, A. Veeraraghavan, and J. Kautz, "Reblur2deblur: Deblurring videos via self-supervised learning," in *Proc. ICCP*, 2018, pp. 1–9.
- [38] T. M. Nimisha, K. Sunil, and A. Rajagopalan, "Unsupervised class-specific deblurring," in *Proc. ECCV*, 2018, pp. 353–369.
- [39] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li, "Deblurring by realistic blurring," in *Proc. CVPR*, 2020, pp. 2737–2746.
- [40] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," in *Proc. CVPR*, 2020, pp. 3341–3350.
- [41] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning," in *Proc. CVPR*, 2021, pp. 9137–9146.
- [42] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.
- [43] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [44] E. P. Simoncelli and E. H. Adelson, "Noise removal via bayesian wavelet coring," in *Proc. ICIP*, vol. 1, 1996, pp. 379–382.
- [45] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. CVPR*, vol. 2, 2005, pp. 60–65.
- [46] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction wiener filter," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 14, no. 4, pp. 1218–1234, 2006.
- [47] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [48] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [49] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. ICCV*, 2009, pp. 2272–2279.
- [50] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [51] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. NeurIPS*, 2012, pp. 341–349.
- [52] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. NeurIPS*, 2016, pp. 2802–2810.
- [53] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, 2016.
- [54] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proc. CVPR*, 2017, pp. 3929–3938.
- [55] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [56] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proc. ICCV*, 2017, pp. 4539–4547.
- [57] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2018.
- [58] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-cnn for image restoration," in *Proc. CVPRW*, 2018, pp. 773–782.
- [59] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. ICCV*, 2019, pp. 3155–3164.
- [60] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided cnn for image denoising," *Neural Networks*, vol. 124, pp. 117–129, 2020.
- [61] W. Liu, Q. Yan, and Y. Zhao, "Densely self-guided wavelet network for image denoising," in *Proc. CVPRW*, 2020, pp. 432–433.
- [62] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," in *Proc. ECCV*, 2020, pp. 492–511.
- [63] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "Nbnet: Noise basis learning for image denoising with subspace projection," in *Proc. CVPR*, 2021, pp. 4896–4906.
- [64] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.
- [65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
- [66] L. Guo, S. Huang, H. Liu, and B. Wen, "Toward robust image denoising via flow-based joint image and noise model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, pp. 6105–6115, 2024.
- [67] Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. V. Gool, "Denoising diffusion models for plug-and-play image restoration," in *Proc. CVPRW*, 2023, pp. 1219–1229.
- [68] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. CVPR*, 2019, pp. 1712–1722.
- [69] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang, "Variational denoising network: Toward blind noise modeling and removal," *Proc. NeurIPS*, vol. 32, pp. 1690–1701, 2019.
- [70] Y. Kim, J. W. Soh, G. Y. Park, and N. I. Cho, "Transfer learning from synthetic to real-noise denoising with adaptive instance normalization," in *Proc. CVPR*, 2020, pp. 3482–3492.
- [71] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," in *Proc. ICML*, 2018, pp. 2965–2974.
- [72] J. Batson and L. Royer, "Noise2self: Blind denoising by self-supervision," in *Proc. ICML*, 2019, pp. 524–533.
- [73] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," in *Proc. CVPR*, 2019, pp. 2129–2137.
- [74] X. Wu, M. Liu, Y. Cao, D. Ren, and W. Zuo, "Unpaired learning of deep image denoising," in *Proc. ECCV*, 2020, pp. 352–368.
- [75] W. Du, H. Chen, and H. Yang, "Learning invariant representation for unsupervised image restoration," in *Proc. CVPR*, 2020, pp. 14483–14492.
- [76] Y. Zhao, L.-M. Po, T. Lin, X. Wang, K. Liu, Y. Zhang, W.-Y. Yu, P. Xian, and J. Xiong, "Legacy photo editing with learned noise prior," in *Proc. WACV*, 2021, pp. 2103–2112.
- [77] Y. Mansour and R. Heckel, "Zero-shot noise2noise: Efficient image denoising without any data," in *Proc. CVPR*, 2023, pp. 14018–14027.
- [78] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.
- [79] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," in *Proc. ACM SIGGRAPH*, 2007, pp. 1–es.
- [80] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 168–186, 2012.
- [81] H. Yamashita, D. Sugimura, and T. Hamamoto, "Rgb-nir imaging with exposure bracketing for joint denoising and deblurring of low-light color images," in *Proc. ICASSP*, 2017, pp. 6055–6059.
- [82] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. ICCV*, 2017, pp. 764–773.
- [83] W. Shen, M. Cheng, G. Lu, G. Zhai, L. Chen, M. S. Asif, and Z. Gao, "Spatial temporal video enhancement using alternating exposures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 4912–4926, 2021.
- [84] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. CVPR*, 2018, pp. 3291–3300.
- [85] C.-H. Liang, Y.-A. Chen, Y.-C. Liu, and W. Hsu, "Raw image deblurring," *IEEE Trans. Multimedia*, vol. 24, pp. 61–72, 2022.
- [86] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. on Graphics*, vol. 35, no. 6, pp. 1–12, 2016.
- [87] Y. Zhao, L.-M. Po, T. Zhang, Z. Liao, X. Shi *et al.*, "Saliency map-aided generative adversarial network for raw to rgb mapping," in *Proc. ICCVW*, 2019, pp. 3449–3457.

- [88] L. Liu, X. Jia, J. Liu, and Q. Tian, "Joint demosaicing and denoising with self guidance," in *Proc. CVPR*, 2020, pp. 2240–2249.
- [89] W. Xing and K. Egiazarian, "End-to-end learning for joint image demosaicing, denoising and super-resolution," in *Proc. CVPR*, 2021, pp. 3507–3516.
- [90] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.
- [91] S. Gu, Y. Li, L. V. Gool, and R. Timofte, "Self-guided network for fast image denoising," in *Proc. ICCV*, 2019, pp. 2511–2520.
- [92] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 132–141, 2006.
- [93] W. Zhang, G. Zhai, Y. Wei, X. Yang, and K. Ma, "Blind image quality assessment via vision-language correspondence: A multitask learning perspective," in *Proc. CVPR*, 2023, pp. 14071–14081.
- [94] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
- [95] "Pytorch-opcounter," <https://github.com/Lyken17/pytorch-OpCounter>, accessed: Sep. 30, 2021. [Online].
- [96] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "Hinet: Half instance normalization network for image restoration," in *Proc. CVPRW*, 2021, pp. 182–192.



**Xin Ye** received the B.S. degree in physics from Zhejiang University in 2014 and received the Ph.D. degree in physics from the Chinese University of Hong Kong in 2019. His research interests include image processing and deep learning.



**Yuzhi Zhao** (S'19) received the B.Eng. degree in Electronic and Information Engineering from Huazhong University of Science and Technology (HUST) in 2018, and the Ph.D. degree in Electronic Engineering from City University of Hong Kong (CityU) in 2023.

His research interests include low-level vision, computational photography, and generative models (such as diffusion models and multimodal large language models). He has authored 10 first-authored works and has contributed to over 35 papers in prestigious venues. He also serves as a peer reviewer for international conferences and journals, including CVPR, ICCV, ECCV, SIGGRAPH, AAAI, PR, IJCV, CVIU, IEEE TMM, IEEE TCSV, IEEE TNNLS, IEEE TIP, ACM TOG, etc.



**Yongzhe Xu** received the B.Eng. and M.Eng. degrees in software engineering from Sun Yat-sen University, Guangzhou, China, in 2017 and 2019, respectively. He is currently an algorithm engineer with the Computational Imaging Group at Tetras.ai. His research interests include generative models, computational photography , and deep learning.



**Lai-Man Po** (M'92–SM'09) received the B.S. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Hong Kong, in 1988 and 1991, respectively. He has been with the Department of Electronic Engineering, City University of Hong Kong, since 1991, where he is currently an Associate Professor of Department of Electrical Engineering. He has authored over 150 technical journal and conference papers. His research interests include image and video coding with an emphasis deep learning based computer vision algorithms.

Dr. Po is a member of the Technical Committee on Multimedia Systems and Applications and the IEEE Circuits and Systems Society. He was the Chairperson of the IEEE Signal Processing Hong Kong Chapter in 2012 and 2013. He was an Associate Editor of HKIE Transactions in 2011 to 2013. He also served on the Organizing Committee, of the IEEE International Conference on Acoustics, Speech and Signal Processing in 2003, and the IEEE International Conference on Image Processing in 2010.



**Qiong Yan** received her Ph.D. degree in computer science and engineering from Chinese University of Hong Kong in 2013 and the Bachelor's degree in computer science and technology from University of Science and Technology of China in 2009. She is now a research director in SenseTime, leading a group on computational imaging related research and production. Her research focuses on low-level vision tasks, such as image/video restoration and enhancement, image editing and generation.