A Decoupled Learning Scheme for Real-world Burst Denoising from Raw Images

Zhetong Liang[1,2], Shi Guo[1,2], Hong Gu[3], Huaqi Zhang[3], and Lei Zhang[1,2] *

[1] Department of Computing, The Hong Kong Polytechnic University
[2] DAMO Academy, Alibaba Group
[3] vivo Mobile Communication Co., Ltd
{csztliang, csshiguo, cslzhang}@comp.polyu.edu.hk
{guhong, zhanghuaqi}@vivo.com

Abstract. The recently developed burst denoising approach, which reduces noise by using multiple frames captured in a short time, has demonstrated much better denoising performance than its single-frame counterparts. However, existing learning based burst denoising methods are limited by two factors. On one hand, most of the models are trained on video sequences with synthetic noise. When applied to real-world raw image sequences, visual artifacts often appear due to the different noise statistics. On the other hand, there lacks a real-world burst denoising benchmark of dynamic scenes because the generation of clean ground-truth is very difficult due to the presence of object motions. In this paper, a novel multi-frame CNN model is carefully designed, which decouples the learning of motion from the learning of noise statistics. Consequently, an alternating learning algorithm is developed to learn how to align adjacent frames from a synthetic noisy video dataset, and learn to adapt to the raw noise statistics from real-world noisy datasets of static scenes. Finally, the trained model can be applied to real-world dynamic sequences for burst denoising. Extensive experiments on both synthetic video datasets and real-world dynamic sequences demonstrate the leading burst denoising performance of our proposed method.

Keywords: Burst denoising, real-world image denoising, convolutional neural networks, decoupled learning

1 Introduction

The imaging quality of smartphone cameras is much affected by the small aperture and small CMOS sensor, which limit the amount of collected light and result in heavy noise in the raw images. Denoising is a crucial step in the camera image processing pipeline (ISP) to remove the noise and reveal the latent image details. The denoising algorithms can be divided into single-frame denoising methods [12, 39, 17, 3] and burst denoising methods [43, 18, 28, 15]. While the

* Corresponding author. This work is supported by the Hong Kong RGC RIF grant (R5001-18).

former ones take a single-frame image as input for processing and are easier to implement, their denoising performance is limited, especially under the low-light environment. The recently developed burst denoising methods capture multiple frames in a short time as input, and thus they can leverage more redundant information for noise removal, leading to much better denoising quality.

The burst denoising problem can be addressed by hand-crafted methods [18, 11, 25, 12, 43] or learning-based methods [28, 36, 15]. The traditional hand-crafted algorithms are often manually designed to exploit the spatio-temporal similarities. For example, the well-known VBM3D method [11] denoises an image patch by finding and fusing its similar patches in the adjacent frames. In contrast, the learning-based methods train a denoising model by using pairwise datasets with a noisy image sequence as input and a clean image as ground-truth. In particular, the rapid development of deep convolutional neural networks (CNNs) [28, 36, 15] largely facilitate the research of learning based burst denoising. The CNN model is powerful to learn a set of nonlinear transformations from the noisy input to the clean output, including frame alignment, fusion and post processing, achieving superior performance to traditional burst denoising methods.

Despite the great progress, the learning-based burst denoising methods are limited by two factors. On one hand, the current multi-frame CNN models are mostly trained on video datasets with synthetic noise, e.g., Gaussian or Poisson-Gaussian noises. When the learned models are applied to real-world raw image sequences, whose noise distribution and statistics are more complex, unpleasant visual artifacts such as color shift and residual noise will appear. One the other hand, there lacks a real-world dataset for learning burst denoising models of dynamic sequence. This is mainly because in the presence of scene motion (e.g., hand shake motion and object motion), it is difficult to craft a clean ground-truth frame by using existing ground-truth generation techniques, such as using low ISO setting [9] or averaging multiple frames [1]. Misalignment problem will occur, which significantly degrades the quality of ground-truth. It is highly desirable to develop a burst denoising CNN model that can adapt to the real-world noise statistics without the need of a real-world pairwise burst image dataset.

There are two key issues in designing such a burst denoising CNN model. Firstly, to enable multi-frame processing, the CNN model should be able to align input frames to compensate the scene motion caused by hand shake and object movement in real scenarios. Second, the CNN model should be able to adapt to real-world noise for better generalization to real-world burst images. Based on the above considerations, in this paper we propose a decoupled learning framework for real-world burst denoising. First, a novel multi-frame CNN model is carefully designed with modular architecture which decouples the learning of motion from the learning of noise adaption. Second, an alternative learning algorithm is developed to leverage the complementary information from two datasets we prepared. One is a video dataset with synthetic noise, where the model learns to perform frame alignment, while the other is a real-world burst image dataset of static scenes, from which the model learns to adapt to raw noise statistics. With the designed CNN model and our decoupled learning algorithm, the learned

CNN model achieves leading performance in real-world burst denoising without the need of a pairwise real-world burst dataset for training.

The rest of the paper is organized as follows. Section 2 reviews some related work. Section 3 describes in detail the proposed decoupled learning method. Section 4 presents the experimental results and Section 5 concludes the paper.

2 Related Work

2.1 Synthetic Image Denoising

Many image denoising algorithms are developed and evaluated on the images corrupted by synthetic noise. At early stage, prior knowledge of natural images is exploited for denoising, including statistical prior [32, 30], sparsity prior [2, 14, 27] and non-local self-similarity [7, 12, 16, 35, 26]. The performance of these traditional methods is limited because the hand-crafted priors are not strong enough to characterize the complex structures of natural images. Recently, deep learning-based approaches have been developed for denoising tasks with substantial progress [39, 40, 21, 37, 3, 33, 24]. The DnCNN model showed that a CNN-based method can outperform non-CNN denoising methods by a large margin [39]. Recently, Anwar *et al.* introduced the feature attention operation into the denoising CNN model, achieving state-of-the-art image quality [3]. Other representative works include FFDNet [40], MemNet [33], MWCNN [24], etc.

2.2 Real-world Image Denoising

The research on real-world image restoration has not been fully conducted until recently owe to the several real-world datasets constructed for this purpose [8, 41, 29, 1, 9]. For the task of real-world denoising, Plotz *et al.* established a benchmark [29], where a pairwise dataset is collected by taking high/low ISO images. Abdelhamed *et al.* built a dataset of static scenes collected by smartphone cameras [1]. Each data pair is composed of a sequence of noisy raw images and the corresponding clean ground-truth image created by frame averaging. Chen *et al.* collected an image dataset [9] and a video dataset [10] by using high/low ISO settings to capture static noisy/clean raw images in low-light environment.

In addition to these real-world datasets, several works have been reported to synthesize realistic data for denoising [5, 17, 28, 13]. Tim *et al.* [5] and Guo *et al.* [17] proposed to reverse the ISP pipeline on the sRGB images and generate noisy training images that are close to the camera raw data. However, these methods are compromised schemes which cannot cope with the real-world scenes with heavy noise corruption and object motion.

2.3 Burst Denoising

Burst denoising methods, an advantage over single-frame ones, take a noisy image sequence as input, and perform a series of operations, including frame

alignment, temporal fusion and post-processing, to reproduce the underlying scene [43, 18, 11, 25, 23, 6]. The frame alignment operation aims to build the correspondence between the dynamic contents of the target and reference frames. Some works adopt block matching for alignment [18, 11, 25, 12, 43], while others use optical flow methods [23, 6]. The fusion operation aims to merge the outputs from multiple frames, which should be robust to alignment error. Representative approaches include collaborative filtering [12], non-local means [6] and frequency domain fusion [18].

Recently, a few works have been proposed to learn frame alignment and fusion from the input sequences for burst denoising. The KPN model [28] predicts the convolutional kernels to selectively fuse a burst of images with object motion. Xue *et al.* [36] designed a CNN model that explicitly consists of frame alignment, fusion and post-processing modules. Godard *et al.* [15] proposed a recurrent architecture for burst denoising, which can increase the image quality by accumulating noisy images. These learning-based methods achieve better image quality than their non-learning counterparts.

3 Decoupled Learning Network for Burst Denoising

3.1 Problem Statement

Given a sequence of N noisy raw images (e.g., in the Bayer color filter array (CFA) pattern [4]) captured by a handheld camera, denoted by $\boldsymbol{I} = \{I_1, I_2, ..., I_N\}$, our goal is to estimate a clean RGB image O from $\boldsymbol{I}$, i.e., $O = f(\boldsymbol{I}; \theta)$, where $f(\cdot; \theta)$ denotes the denoising model (e.g., a CNN model in our work) parameterized by θ. We consider one frame from $\boldsymbol{I}$ as the reference frame, denoted by I_r, and denoise it by aligning and fusing it with other frames $I_i, i \neq r$.

To denoise real-world burst image sequences of dynamic scenes, the CNN model should learn to simultaneously align frames and adapt to real-world noise from some training dataset. Considering the fact that there lacks a real-world burst image dataset of dynamic scenes with ground-truth clean images, we propose to use two types of datasets for training, which can be generated by using the publically accessible data. One is a synthetic noisy video dataset of dynamic scenes, denoted by $\mathcal{D}_d$ (subscript "d" for dynamic). Each data pair ($\boldsymbol{I}_d$,G_d) in $\mathcal{D}_d$ consists of a noisy video sequence $\boldsymbol{I}_d$ and a clean ground-truth frame G_d. The other is a real-world burst image dataset of static scenes, denoted by $\mathcal{D}_s$ (subscript "s" for static). Each data pair ($\boldsymbol{I}_s$,G_s) in $\mathcal{D}_s$ consists of a noisy raw image sequence $\boldsymbol{I}_s$ and a ground-truth clean RGB image G_s.

$\mathcal{D}_d$ can be easily built by using the many high quality video sequences [36], while $\mathcal{D}_s$ can be built by the existing frame averaging method [1]. These two datasets have complementary information. The video dataset $\mathcal{D}_d$ contains rich dynamic scene motions, but the noise is synthetic and not real. In contrast, the static burst dataset $\mathcal{D}_s$ does not contain scene motion, but can provide information of real noise statistics. In this paper, we investigate how to learn a CNN model $f(\cdot; \theta)$ from $\mathcal{D}_d$ and $\mathcal{D}_s$, and present a decoupled learning scheme to achieve this goal.

3.2 Datasets Preparation

Before we present the CNN model architecture and the decoupled learning scheme, the two required datasets, $\mathcal{D}_d$ and $\mathcal{D}_s$, must be prepared. We present how to use the existing data to build these two datasets in this section.

Preparation of $\mathcal{D}_d$. We collect high quality video sequences from some video dataset (e.g., Vimeo-90k dataset [36]) to prepare $\mathcal{D}_d$. Specifically, every N consecutive frames are extracted as a burst sequence from the videos. However, directly adding noise to those sequences will make $\mathcal{D}_d$ deviate too much from the real-world dynamic noisy image sequences. Inspired by the work of [5], we propose to reverse the ISP pipeline and add noise to the reversed raw images so that the synthesized noisy sequences can be more realistic.

Specifically, we reverse four key ISP operations, including gamma correction, color space conversion, white balance and demosaicking, together with realistic noise synthesis, for building $\mathcal{D}_d$. A reverse gamma conversion with parameter γ is applied on a video frame L, where γ is sampled from a uniform distribution within range [2.0,2.6]. Then, a reverse color space conversion C is applied, with the color matrix randomly interpolated by the color matrices given in static real-world dataset $\mathcal{D}_s$. Next, a reverse white balance gain of $W = 1/(r_g, 1, b_g)$ is applied with r_g and b_g matched to the statistics in $\mathcal{D}_s$. Finally, we obtain the synthetic clean RGB image of a frame as $G = WCL^{\gamma}$.

To synthesize the noisy input, a mosaicking mask M is applied to G, yielding a Bayer CFA pattern image, denoted by G_M. Then Poisson-Gaussian noise which is approximated by heteroscedastic Gaussian [28] is added to the CFA image to synthesize noisy raw image I:

$$I = G_M + n(G_M) \tag{1}$$

where noise n is dependent on the signal intensity g at each location:

$$n(g) \sim \mathcal{N}(\mu = g, \sigma^2 = \lambda_{shot} g + \lambda_{read}^2) \tag{2}$$

where $\mathcal{N}(\mu, \sigma^2)$ is Gaussian distribution. λ_{shot} and λ_{read} are the shot noise and readout noise, which are uniformly sampled in the range (0.00001,0.01) and (0,0.058), respectively.

By the above described process, we can synthesize a sequence of noisy raw images I and take them as $\boldsymbol{I}_d$. The clean RGB image G of the center frame is taken as the ground-truth G_d. A data pair ($\boldsymbol{I}_d$,G_d) is then constructed for $\mathcal{D}_d$.

Preparation of $\mathcal{D}_s$. We use the static burst image datasets in [9, 1] to prepare dataset $\mathcal{D}_s$. We extract 140 and 162 groups of data pairs in [9] and [1], respectively. Each group contains a static noisy sequence of 5 raw images and a clean RGB ground-truth. We propose to add simple motions to the static burst sequences to facilitate the learning of frame alignment. Specifically, for a raw noisy image sequence, we add vertical and horizontal global shifts to its frames (except for its reference frame I_r):

$$\hat{I}_i = I_i(x + x_i, y + y_i), \quad for \quad i \neq r \tag{3}$$

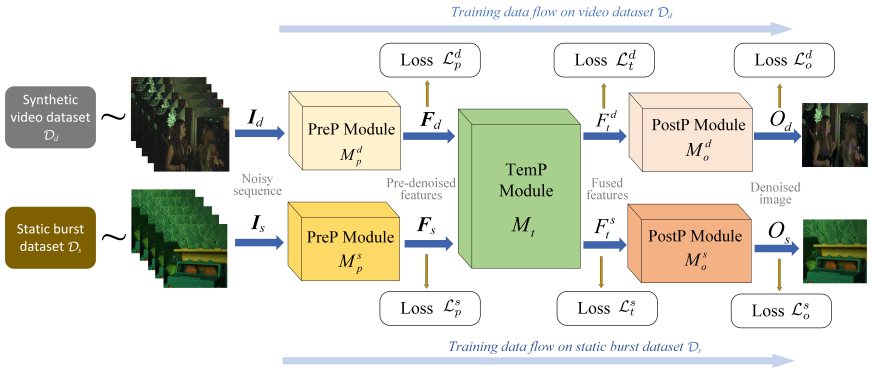
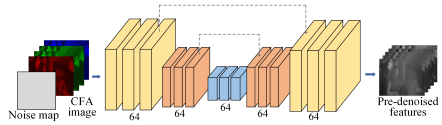
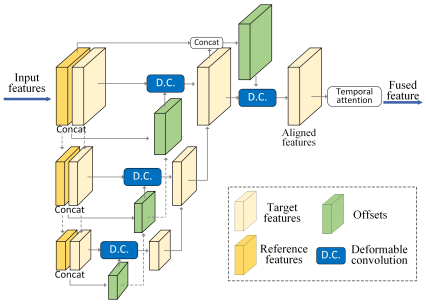


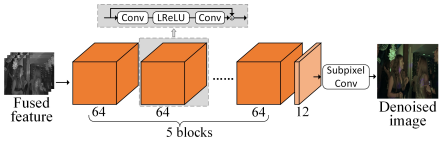
Fig. 1: The decoupled learning framework for our burst denoising network (BDNet).



(a) PreP module



(b) TemP module



(c) PostP module

Fig. 2: The structure of the PreP module M_p, TemP module M_t and PostP module M_o of the proposed BDNet.

where the shift x_i and y_i are uniformly sampled from the range [-4,4].

The ground-truth image G_s is already available in the static noisy image datasets [9, 1]. After adding simple motions to its adjacent noisy frames and taking them as $\boldsymbol{I}_s$, a data pair ($\boldsymbol{I}_s$,G_s) for the dataset D_s can be generated.

3.3 Decoupled Network Design

To achieve the goal of decoupled learning with $\mathcal{D}_d$ and $\mathcal{D}_s$, we design a modular CNN which is explicitly divided into a pre-processing (PreP) module M_p, a temporal processing module (TemP) M_t and a post-processing module (PostP) M_o. We call the proposed CNN model BDNet (burst denoising network), whose learning framework is illustrated in Fig. 1. The detailed structures of modules M_p, M_t and M_o are illustrated in Fig. 2.

Pre-processing module. The PreP module M_p is constructed to perform single-frame denoising on the noisy CFA sequence $\boldsymbol{I} = \{I_1, I_2, ..., I_N\}$ and output

pre-denoised features $\boldsymbol{F} = \{F_1, F_2, ..., F_N\}$. In addition, we add a noise level as input, which is obtained by $\sqrt{\lambda_{shot} + \lambda_{read}^2}$. We adopt a multi-scale (three scales) UNet [31] with 15 convolutional layers for single image denoising for its simplicity and good performance. As shown in Fig. 2(a), the adopted UNet consists of a contracting path which continuously downsamples the image features with stride convolutions, and an expanding path that gradually upsamples the features to the original resolution. Skip connections are added between the contracting and expanding paths at the same scale level. The PreP module M_p not only helps to reduce the noise but also increases the robustness in the subsequent frame alignment operation.

Temporal processing module. The TemP module M_t is constructed to align and fuse the pre-denoised features $\boldsymbol{F} = \{F_1, F_2, ..., F_N\}$ and output a single feature map F_t. It has been shown that accurate alignment can be obtained with deformable convolutions [34]. Thus, we adopt the Pyramid, Cascading and Deformable alignment (PCD) model and temporal attention methods in [34] as the alignment and fusion components in our TemP module, respectively. As shown in Fig. 1(b), the PCD takes a pair of reference and target features as input, and progressively warps the target feature to the reference feature in a multi-scale and cascading manner. The temporal attention component fuses all the aligned features according to their similarities to the reference feature.

Post-processing module. The PostP module M_o takes the fused feature F_t as input and conducts some refinement operations to reconstruct a clean image. As shown in Fig. 1(c), we deploy 5 residual blocks to build M_o, each containing two convolutional layers. Then a 1×1 convolutional and a sub-pixel convolutional layer are applied to output the denoised RGB image O.

3.4 Decoupled Learning Process

Given the BDNet model in Section 3.3 and the two prepared datasets $\mathcal{D}_d$ and $\mathcal{D}_s$ in Section 3.2, the remaining question is how to effectively learn frame alignment and real-world noise adaptation for burst denoising. We propose a decoupled learning method to this end, which is illustrated in Fig. 1.

First, considering that the noise statistics in the dynamic video dataset $\mathcal{D}_d$ (synthetic noise) and static burst dataset $\mathcal{D}_s$ (real-world noise) are different, different CNN modules should be deployed for each case to avoid mixed learning. Therefore, we train and deploy two instances of the PreP module M_p with the same architecture but different parameters. These two module instances, denoted by M_p^d and M_p^s, transform the synthetic noisy sequences $\boldsymbol{I}_d$ (from $\mathcal{D}_d$) and real-world noisy sequences $\boldsymbol{I}_s$ (from $\mathcal{D}_s$) to pre-denoised feature sequences $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$, respectively. We assign a pair of sub-losses, denoted by $\mathcal{L}_p^d$ and $\mathcal{L}_p^s$, for the pre-denoising modules

$$\begin{cases} \min \mathcal{L}_p^d\big(G_d, Recon_1(F_{d,r})\big) \\ \min \mathcal{L}_p^s\big(G_s, Recon_1(F_{s,r})\big) \end{cases} \tag{4}$$

where $F_{d,r}$ and $F_{s,r}$ are the reference feature maps in the pre-denoised feature sequences $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$, respectively. This pair of sub-losses $\mathcal{L}_p^d$ and $\mathcal{L}_p^s$ (e.g., ℓ_1

loss) calculate the errors between the ground-truths G_d, G_s from the two datasets and the images reconstructed from the pre-denoised reference features $F_{d,r}$, $F_{s,r}$, respectively. The reconstruction operation $Recon_1$ is performed by a shared 1×1 convolution that reduces the channel size, followed by a sub-pixel convolution to expand to the original resolution. Since the features are initially denoised, they are in a relatively clean signal space, which facilitate the subsequent frame alignment learning.

Second, we deploy one TemP module M_t to receive the feature sequences $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$, perform frame alignment and fusion, and output the fused features F_t^d and F_t^s, respectively. Since both $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$ are in a relatively clean latent space, the learned frame alignment capability of $\boldsymbol{F}_d$ can be transfered to $\boldsymbol{F}_s$. A pair of sub-losses, denoted by $\mathcal{L}_t^d$ and $\mathcal{L}_t^s$, are deployed on M_t:

$$\begin{cases} \min \mathcal{L}_t^d\big(G_d, Recon_2(F_t^d)\big) \\ \min \mathcal{L}_t^s\big(G_s, Recon_2(F_t^s)\big) \end{cases} \tag{5}$$

The sub-losses compare the ground-truths G_d and G_s with the images reconstructed from the fused features F_t^d and F_t^s, respectively. The reconstruction operation $Recon_2$ consists of a shared 1×1 convolution followed by a sub-pixel convolution.

Third, considering that the dataset $\mathcal{D}_d$ is generated by reversing the ISP, while the images in dataset $\mathcal{D}_s$ are collected in the real raw image domain, the ground-truth images of the two datasets may have some appearance differences. In particular, the ground-truth images in $\mathcal{D}_s$ have genuine image structures, whereas the ones in $\mathcal{D}_d$ may have artifacts caused by reversing ISP. Therefore, different CNN modules should be deployed to learn different types of ground-truths. We assign two instances of PostP module M_o, denoted by M_o^d and M_o^s, to transform the fused features F_t^d and F_t^s to the final denoised images O_d and O_s, respectively. A pair of sub-losses, denoted by $\mathcal{L}_o^d$ and $\mathcal{L}_o^s$, are deployed to compare G_d and G_s with the denoised images O_d and O_s, respectively:

$$\begin{cases} \min \mathcal{L}_o^d\big(G_d, O_d)\big) \\ \min \mathcal{L}_o^s\big(G_s, O_s)\big) \end{cases} \tag{6}$$

Finally, in the training process, we have two sets of loss functions $\mathcal{L}^d$ and $\mathcal{L}^s$ to update the BDNet on $\mathcal{D}_d$ and $\mathcal{D}_s$, respectively, which are as follows:

$$\begin{cases} \mathcal{L}^d = w_p(k) \cdot \mathcal{L}_p^d + w_t(k) \cdot \mathcal{L}_t^d + w_o(k) \cdot \mathcal{L}_o^d \\ \mathcal{L}^s = w_p(k) \cdot \mathcal{L}_p^s + w_t(k) \cdot \mathcal{L}_t^s + w_o(k) \cdot \mathcal{L}_o^s \end{cases} \tag{7}$$

where $w_p(k)$, $w_t(k)$ and $w_o(k)$ are the weights assigned on the sub-losses, which are variables dependent on the global epochs k in the training. We adopt an adaptive weighting scheme to train the modules progressively by setting:

$$\begin{cases} w_p(k) = 0.1^{\frac{k}{K}}, & 1 \le k \le K, \quad else \quad 0.1 \\ w_t(k) = 0.1 \cdot 10^{\frac{k-K}{K}}, & K \le k \le 2K, \quad else \quad 0.1 \\ w_o(k) = 0.1 \cdot 10^{\frac{k-2K}{K}}, & 2K \le k \le 3K, \quad else \quad 0.1 \end{cases} \tag{8}$$

Under this weighting scheme, the three pairs of sub-losses in Eq. (7) dominate the training process in turn. In the first K epochs, $w_p(k)$ gradually decreases from 1 to 0.1, while the others remain at 0.1. This setting emphasizes the sub-losses $\mathcal{L}_p^d$ and $\mathcal{L}_p^s$ that optimize the PreP module. Then, during the epochs from K to $2K$, the weight $w_t(k)$ gradually ascends from 0.1 to 1, with the others remain at 0.1. At this stage, the sub-loss $\mathcal{L}_t^d$ and $\mathcal{L}_t^s$ dominate the training, focusing on the TemP module. Lastly, during the epoch from $2K$ to $3K$, the weight $w_o(k)$ on sub-losses $\mathcal{L}_o^d$ and $\mathcal{L}_o^s$ ascends from 0.1 to 1, with the other weights remaining at 0.1. This stage focuses on the training of the PostP module.

We adopt ℓ_1 loss for all the sub-losses involved in Eq. (7). An alternative training scheme is adopted to assign J_1 iterations for loss $\mathcal{L}^d$ and J_2 iterations for loss $\mathcal{L}^s$ in one cycle. In the testing stage, the modules M_p^d and M_o^d are removed, and only the M_p^s, M_t and M_o^s modules are used to form the final BDNet model.

4 Experiments

In this section, we conduct experiments to verify the effectiveness of proposed decoupled learning approach for burst denoising. We evaluate our BDNet on both synthetic noisy video dataset and real-world noisy sequences quantitatively and qualitatively. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [42] are used as the quantitative metrics.

The kernel size of the convolutional layers of our BDNet is set to 3×3. Leaky ReLU is used as the activation function. The number of input frames N of a burst sequence is set to 5 for all multi-frame methods in the comparison. In all experiments, we use the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.99$) [19] to train BDNet and other competing CNN models. The initial learning rate is set to 10^{-4}, and it exponentially decays by 0.1 at 3/4 of the total epochs. The parameter K in Eq. (8) is set to 30. In the decoupled training, we update the model for $J_1 = 3$ iterations on $\mathcal{D}_d$ and $J_2 = 1$ on $\mathcal{D}_s$ in one cycle. In all training, the batch size is set to 2 and the patch size is set to 128×128. Random rotations, vertical and horizontal flippings are applied for data augmentation.

4.1 Datasets

Training set. For dynamic video dataset $\mathcal{D}_d$, we extract 20,000 image sequences from the Vimeo-90K video dataset [36], each containing 5 consecutive frames. As for $\mathcal{D}_s$, we leverage the SIDD [1] and SID datasets [9] to build it for multi-camera training since none of the two datasets has enough training data for a single camera. Specifically, we combine the Sony training set of SID (162 image sequences) and 140 image sequences selected from SIDD training set as our static burst dataset $\mathcal{D}_s$.

Testing set. Our testing set consists of a synthetic test set and a real-world test set. For the synthetic test set, we extract another 200 image sequences (different from the training sequences in scene and content) from the Vimeo-90k dataset [36], denoted by Vimeo-200. For the real-world test set, we build a static

Table 1: Quantitative results (PSNR/SSIM) on the synthetic test sets. G25, G50 and PG indicates Gaussian σ=25, Gaussian σ=50 and Poisson-Gaussian noise, respectively.

	VBM4D	DNCNN	RIDNet	KPN	TOFlow	BDNet
G25	28.30/0.735	32.60/0.870	34.74/0.908	34.84/0.907	34.99/0.902	36.78/0.937
G50	25.92/0.621	29.32/0.776	31.47/0.821	32.44/0.862	31.95/0.829	34.03/0.900
PG	30.48/0.845	35.79/0.934	38.34/0.954	37.77/0.940	37.90/0.951	39.45/0.965

test set, denoted by Real-static, for quantitative evaluation, as well as a dynamic test set, denoted by Real-dynamic, for qualitative perceptual evaluation because the ground-truths are hard to generate for dynamic scenes. The Real-static set is composed of the Sony test set (50 image sequences) in SID dataset [9] and 20 image sequences selected from the SIDD dataset [1]. For the Real-dynamic test set, we use iPhone 7 to capture 20 dynamic noisy image sequences in low-light environment. All the images are stored in raw format.

4.2 Results on Synthetic Noisy Sequences

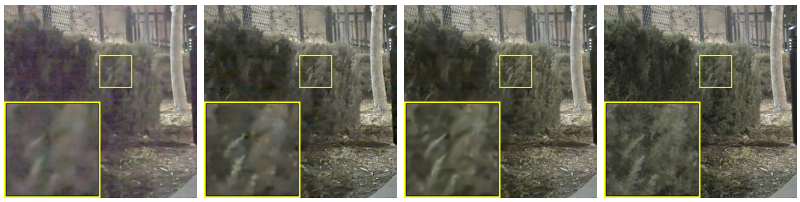
We firstly evaluate the burst denoising performance of our BDNet on synthetic noisy data. We compare BDNet with several representative and state-of-the-art methods which are popularly used for synthetic noisy video denoising, including VBM4D [26], DnCNN [39], RIDNet [3], KPN [28] and TOFlow [36]. Among them, VBM4D is a classical patch based video denoising method; DnCNN and RIDNet are single-frame denoising CNN models; and KPN and TOFlow are CNN based multi-frame denoising models. We train all the CNN based models, including BDNet, until convergence on the dataset $\mathcal{D}_d$. We add three types of noises, including Gaussian noise with σ=25 (G25), Gaussian noise with σ=50 (G50) and Poisson-Gaussian noise (PG) defined in Eq. (2), to the Vimeo-200 test set, and apply the competing models to these synthetic noisy sequences.

Table 1 shows the PSNR/SSIM results of the compared methods. We can see that the proposed BDNet achieves the highest PSNR and SSIM scores in all cases. While TOFlow performs well in the cases of low noise levels, i.e., G25 and PG, its performance heavily degrades in the case of higher noise level, i.e., G50. This is because it performs frame alignment in the image domain, but the alignment accuracy is affected by the heavy image noise. While the single-frame models, DnCNN and RIDNet, have relatively lower PSNR/SSIM scores, RIDNet performs well on PG noise, which may be attributed to its robust feature attention modules. For the visual comparison of the denoising results, the reader can refer to the **supplementary file** for details.

4.3 Results on Real-world Noisy Sequences

We use the "Real-static" (for quantitative evaluation) and "Real-dynamic" (for qualitative evaluation) test sets to evaluate the performance of BDNet on real-

(a) Noisy frame (b) UTR (c) M-UNet (d) M-RIDNet



(e) KPN (f) INN (g) BDNet (h) ground-truth

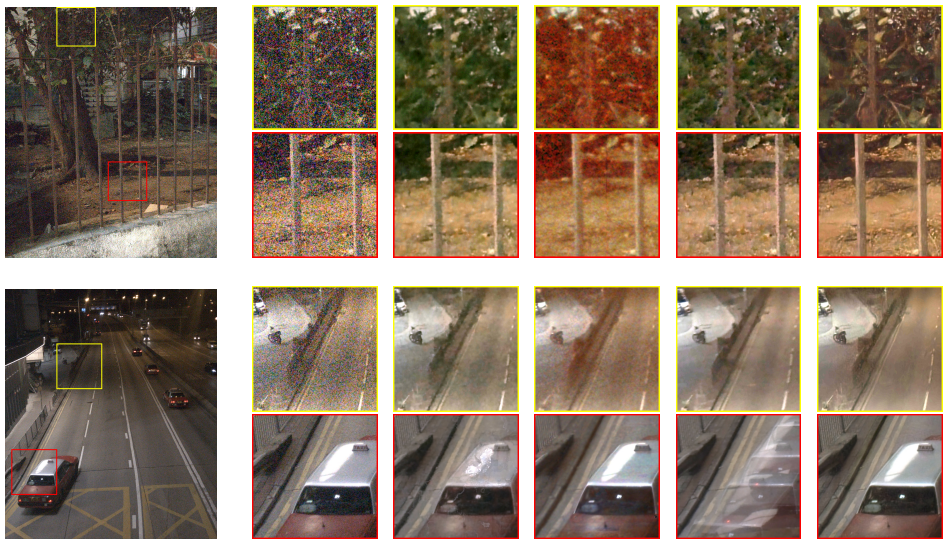
Fig. 3: Denoising Results of the compared methods on Real-static test set. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization.

Table 2: Quantitative evaluation on the Real-static test set.

	VBM4D	UTR	UNet	M-UNet	RIDNet	M-RIDNet	KPN	INN	BDNet
PSNR	40.49	42.02	43.85	44.23	44.17	44.54	39.68	43.95	45.31
SSIM	0.901	0.897	0.954	0.964	0.960	0.968	0.867	0.964	0.971

world burst noisy sequences. We compare BDNet with those methods popularly used for real-world image denoising in literature, including VBM4D [26], UNet [9], RIDNet [3], Unprocess-to-raw (UTR) [5], KPN [28] and INN [20]. Both UTR and KPN methods learn real-world denoising by synthesizing data that resemble raw noisy images. In particular, UTR reverses the ISP pipeline, while KPN adds motion and noise to clean images to synthesize a burst of noisy images. In addition, the INN method use global affine transformation to align frames and performs burst denoising by learning a trainable proximal operation. For fair comparison, we make the following configurations.

1) First, for the single-frame denoising methods UNet and RIDNet, we build a multi-frame version for them, denoted by M-UNet and M-RIDNet, respectively. M-UNet and M-RIDNet first denoise each frame in the noisy sequence, and then apply optical flow alignment [38] to fuse the denoised frames by average fusion, resulting in the finally denoised sequences.
2) Second, the UTR method learns a single-frame CNN. For fair comparison with UTR, we replace its single-frame CNN by our multi-frame BDNet structure and re-train it on $\mathcal{D}_d$.



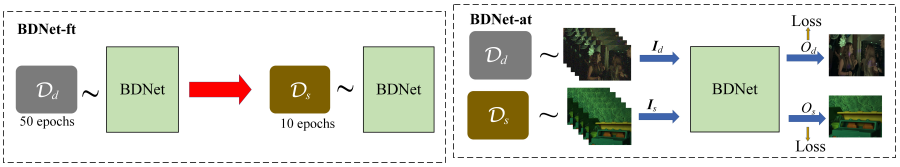
(a) (b) (c) (d) (e) (f)

Fig. 4: Denoising results of the compared methods on Real-dynamic test set. (a) Noisy reference frame. (b) Noisy patches. (c) M-RIDNet. (d) KPN. (e) INN. (f) BDNet. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. Best viewed on screen with zoom-in.

3) Third, we re-train KPN on dataset $\mathcal{D}_d$ using the same data synthesis setting as the original paper [28], including ISP pipeline reversing and noise generation.
4) At last, we train UNet, RIDNet and INN models on dataset $\mathcal{D}_s$ until convergence, and use the models with the best testing performance.

Table 2 shows the quantitative evaluation results on the "Real-static" test set. It is clear that the proposed BDNet obtains the highest PSNR and SSIM scores. UTR and KPN have low objective scores since they are not able to adapt to the real-world static test data. The two multi-frame models, M-UNet and M-RIDNet, obtain higher scores than their single-frame counterparts, which proves that the multi-frame fusion helps for realistic noise removal. However, their PSNR/SSIM results are still lower than the proposed BDNet. Fig. 3 compares visually the denoising results of the compared methods on one image in the Real-static test set. One can see that the proposed BDNet is able to remove the noise without blurring the details, whereas the other methods tend to over-smooth the image details. In addition, the UTR method leaves residual noise in the image (Fig. 3(b)) because it is not adapted to the real-world dataset.

We then compare the competing models on the Real-dynamic test set. Since no ground-truths are available, we can only make qualitative comparisons on them. Fig. 4 shows the results, where we can see that those competing methods



(a) Train on $\mathcal{D}_d$ and fine-tune on on $\mathcal{D}_s$ (b) Direct alternating training

Fig. 5: Illustration of different learning schemes for real-world burst denoising with dynamic scenes. Please refer to the text for detailed descriptions.

Table 3: Quantitative results (PSNR/SSIM) of different learning schemes on the Real-static test set.

BDNet-ft	BDNet-at	Default setting
45.17/0.968	44.67/0.967	**45.31/0.971**

have residual noise or artifacts caused by scene motion. In particular, the KPN method has severe color shift on image with large noise (the plant area in Fig. 4(d)). The M-RIDNet and INN methods encounter motion artifacts in the car area in Fig. 4(c)(e)). This is because optical flow and global affine alignment cannot effectively count for the local object motion. In contrast, the proposed BDNet is able to compensate for scene motion and restore the clean details. More visual comparison results can be found in the **supplementary file**.

4.4 Ablation Study

To better validate the effectiveness of our decoupled learning strategy, we make some ablation studies here by comparing it with two other intuitive training strategies using $\mathcal{D}_d$ and $\mathcal{D}_s$, which are illustrated in Fig. 5. The first scheme, denoted by BDNet-ft, trains BDNet on dataset $\mathcal{D}_d$ and fine-tunes it on $\mathcal{D}_s$ till convergence. The second scheme, denoted by BDNet-at, directly alternates the training on $\mathcal{D}_d$ and $\mathcal{D}_s$ without deploying two instances of the PreP module M_p and the PostP module M_o.

Table 3 shows the quantitative results of the compared schemes on the Real-static test set. It can be seen that BDNet-at has much lower PSNR/SSIM scores than BDNet, which validates the importance of using two instances for M_p and M_o. BDNet-ft achieves similar PSNR/SSIM scores to BDNet. This is mainly because it utilizes $\mathcal{D}_s$ in the training while this quantitative test is also on static scenes. However, the perceptual quality of BDNet-ft and BDNet-na is much worse than BDNet for both Real-static and Real-dynamic scenarios. Fig. 6 shows the denoising results of three schemes on a static low-light sequence. One can see that BDNet-ft and BDNet-na generate visual artifacts in the street lamp area due to insufficient adaption to real-world noise. Fig. 7 shows the denoising results on dynamic scenes. It can be seen that BDNet-ft causes ghost artifacts in the

(a) (b) (c) (d) (e)

Fig. 6: The results on a raw image sequence with large noise in Real-static test set by different learning schemes. (a) Noisy patch. (b) BDNet-ft. (c) BDN-at. (d) Default BDNet. (e) Ground-truth.

(a) (b) (c) (d) (e)

Fig. 7: The results on an raw image sequence in Real-dynamic test set by different learning schemes. (a) Noisy reference frame. (b) Noisy patch. (c) BDNet-ft. (d) BDN-at. (e) Default BDNet. Best viewed on screen with zoom-in.

car area with large motion, because its fine-tuning on static dataset corrupts the learned alignment ability. In contrast, the decoupled learning scheme can achieve both merits of aligning dynamic sequences and revealing fine details in real-world scenes. More visual comparison results can be found in the **supplementary file**.

5 Conclusion

It is a challenging problem to learn a burst denoising network for real-world dynamic noisy sequences because of the lack of a pairwise training dataset. In this paper, we proposed to leverage two types of existing datasets, a synthetic noisy video dataset and a static real-world burst dataset, to address this issue. We designed a modular CNN model, and proposed a decoupled learning approach, which learns to align adjacent frames from the synthetic video dataset and learns to adapt to raw noise statistics from the static burst dataset. The trained CNN model, namely BDNet, can be well applied to real-world dynamic noisy sequences and it obtains compelling detail reconstruction quality with little motion blur. BDNet achieves leading performance, both quantitatively and qualitatively, on the task of burst image sequence denoising in real-world scenes.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
2. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing **54**(11), 4311–4322 (Nov 2006). https://doi.org/10.1109/TSP.2006.881199
3. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
4. Bayer, B.E.: Color imaging array (Jul 20 1976), uS Patent 3,971,065
5. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
6. Buades, A., Lisani, J., Miladinović, M.: Patch-based video denoising with optical flow estimation. IEEE Transactions on Image Processing **25**(6), 2573–2586 (Jun 2016). https://doi.org/10.1109/TIP.2016.2551639
7. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 60–65. IEEE (2005)
8. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: Toward real-world single image super-resolution: A new benchmark and a new model. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
9. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition. pp. 3291–3300 (Jun 2018). https://doi.org/10.1109/CVPR.2018.00347
10. Chen, C., Chen, Q., Do, M.N., Koltun, V.: Seeing motion in the dark. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
11. Dabov, K., Foi, A., Egiazarian, K.: Video denoising by sparse 3D transform-domain collaborative filtering. In: Proc. 15th European Signal Processing Conf. pp. 145–149 (Sep 2007)
12. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (Aug 2007). https://doi.org/10.1109/TIP.2007.901238
13. Ehret, T., Davy, A., Arias, P., Facciolo, G.: Joint demosaicking and denoising by fine-tuning of bursts of raw images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8868–8877 (2019)
14. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing **15**(12), 3736–3745 (Dec 2006). https://doi.org/10.1109/TIP.2006.881969
15. Godard, C., Matzen, K., Uyttendaele, M.: Deep burst denoising. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 560–577. Springer International Publishing, Cham (2018)
16. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 2862–2869 (Jun 2014). https://doi.org/10.1109/CVPR.2014.366
17. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

18. Hasinoff, S.W., Sharlet, D., Geiss, R., Adams, A., Barron, J.T., Kainz, F., Chen, J., Levoy, M.: Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Trans. Graph. **35**(6), 192:1–192:12 (2016)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Kokkinos, F., Lefkimmiatis, S.: Iterative residual cnns for burst photography applications. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 5929–5938. Computer Vision Foundation / IEEE (2019)
21. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). pp. 5882–5891 (Jul 2017). https://doi.org/10.1109/CVPR.2017.623
22. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 2971–2980. PMLR (2018)
23. Liu, C., Freeman, W.T.: A high-quality video denoising algorithm based on reliable motion estimation. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. pp. 706–719. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
24. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-CNN for image restoration. In: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 886–88609 (Jun 2018). https://doi.org/10.1109/CVPRW.2018.00121
25. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. IEEE Transactions on Image Processing **21**(9), 3952–3966 (Sep 2012). https://doi.org/10.1109/TIP.2012.2199324
26. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. IEEE Transactions on Image Processing **21**(9), 3952–3966 (Sep 2012). https://doi.org/10.1109/TIP.2012.2199324
27. Mairal, J., Bach, F.R., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009. pp. 2272–2279. IEEE Computer Society (2009). https://doi.org/10.1109/ICCV.2009.5459452, https://doi.org/10.1109/ICCV.2009.5459452
28. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition. pp. 2502–2510 (Jun 2018). https://doi.org/10.1109/CVPR.2018.00265
29. Plötz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). pp. 2750–2759 (Jul 2017). https://doi.org/10.1109/CVPR.2017.294
30. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image Processing **12**(11), 1338–1351 (Nov 2003). https://doi.org/10.1109/TIP.2003.818640

31. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
32. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 860–867 vol. 2 (Jun 2005). https://doi.org/10.1109/CVPR.2005.160
33. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: Proc. IEEE Int. Conf. Computer Vision (ICCV). pp. 4549–4557 (Oct 2017). https://doi.org/10.1109/ICCV.2017.486
34. Wang, X., Chan, K.C., Yu, K., Dong, C., Loy, C.C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (June 2019)
35. Xu, J., Zhang, L., Zhang, D., Feng, X.: Multi-channel weighted nuclear norm minimization for real color image denoising. In: Proc. IEEE Int. Conf. Computer Vision (ICCV). pp. 1105–1113 (Oct 2017). https://doi.org/10.1109/ICCV.2017.125
36. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision (IJCV) **127**(8), 1106–1125 (2019)
37. Yang, D., Sun, J.: Bm3d-net: A convolutional neural network for transform-domain collaborative filtering. IEEE Signal Processing Letters **25**(1), 55–59 (Jan 2018). https://doi.org/10.1109/LSP.2017.2768660
38. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-L^1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) Pattern Recognition, 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4713, pp. 214–223. Springer (2007). https://doi.org/10.1007/978-3-540-74936-3_22, https://doi.org/10.1007/978-3-540-74936-3_22
39. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (Jul 2017). https://doi.org/10.1109/TIP.2017.2662206
40. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for CNN-based image denoising. IEEE Transactions on Image Processing **27**(9), 4608–4622 (Sep 2018). https://doi.org/10.1109/TIP.2018.2839891
41. Zhang, X., Chen, Q., Ng, R., Koltun, V.: Zoom to learn, learn to zoom. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
42. and Zhou Wang, Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (Apr 2004). https://doi.org/10.1109/TIP.2003.819861
43. Ziwei Liu, Lu Yuan, X.T.M.U., Sun, J.: Fast burst images denoising. ACM Transactions on Graphics (TOG) **33**(6) (2014)