

# Binarized Low-light Raw Video Enhancement

Gengchen Zhang<sup>1</sup>, Yulun Zhang<sup>2</sup>, Xin Yuan<sup>3</sup>, Ying Fu<sup>1\*</sup>

<sup>1</sup>Beijing Institute of Technology, <sup>2</sup>Shanghai Jiao Tong University, <sup>3</sup>Westlake University

{zhanggengchen, fuying}@bit.edu.cn, yulun100@gmail.com, xylab@westlake.edu.cn

## Abstract

Recently, deep neural networks have achieved excellent performance on low-light raw video enhancement. However, they often come with high computational complexity and large memory costs, which hinder their applications on resource-limited devices. In this paper, we explore the feasibility of applying the extremely compact binary neural network (BNN) to low-light raw video enhancement. Nevertheless, there are two main issues with binarizing video enhancement models. One is how to fuse the temporal information to improve low-light denoising without complex modules. The other is how to narrow the performance gap between binary convolutions with the full precision ones. To address the first issue, we introduce a spatial-temporal shift operation, which is easy-to-binarize and effective. The temporal shift efficiently aggregates the features of neighbor frames and the spatial shift handles the misalignment caused by the large motion in videos. For the second issue, we present a distribution-aware binary convolution, which captures the distribution characteristics of real-valued input and incorporates them into plain binary convolutions to alleviate the degradation in performance. Extensive quantitative and qualitative experiments have shown our high-efficiency binarized low-light raw video enhancement method can attain a promising performance. The code is available at <https://github.com/ying-fu/BRVE>.

## 1. Introduction

Videos captured in low-light environments often suffer from degradations such as severe noise, color distortion, and lack of details. This not only causes poor video aesthetic quality but also impairs the application of video in downstream vision tasks [6, 11, 49]. To enhance low-light videos, there are several hardware-based solutions, including using high ISO, long exposure time, large aperture size, and a flashlight. However, these methods all have their own limitations. For example, a high ISO setting amplifies noise, long exposure time causes motion blur in dynamic scenes, and flashlights

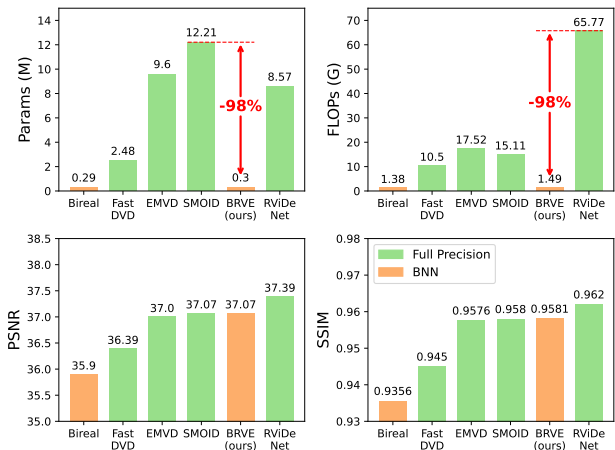


Figure 1. Efficiency and performance comparison of full precision networks and binary neural networks (BNNs).

have a limited range. On the other hand, software-based low-light enhancement methods [4, 5, 9, 14] can compensate for these limitations. Many works [10, 12, 55] have shown raw videos provide several advantages for low-light enhancement, such as linearity to scene luminance and high bit-depth to preserve more dark details. In this work, we follow the raw-to-raw video enhancement pipeline to perform denoising on linearly scaled raw videos.

Deep learning methods have dominated low-light video enhancement [5, 14] and denoising [18, 19, 50, 51]. These deep models, such as convolutional neural networks (CNN) and transformers, often require high computational costs and large memory overhead. Nowadays, using mobile devices (e.g. smartphones and hand-held cameras) to capture videos has become increasingly popular. However, these devices neither have the ability to directly acquire high-quality videos in low-light environments nor do they have enough resources to run deep neural networks. To improve the efficiency of deep neural networks, many network compression techniques are proposed, including network quantization [13, 24], parameter pruning [32, 53], compact network design [52], and knowledge distillation [33]. Among these approaches, binary neural network (BNN) stands out as an extreme case of network quantization, which binarizes

\*Corresponding Author

both weights and activations (features) into only 1-bit (*i.e.*-1 and +1) to reduce the memory overhead and uses bitwise operations to accelerate the computation.

Despite its superiority in efficiency, applying BNN to low-light video enhancement faces several challenges. *i*) It is difficult for BNNs to ensure temporal consistency [5, 48] to avoid flickering and effectively leverage spatial-temporal self-similarity, which is crucial for denoising [1, 22]. Some full precision methods use optical flow [19, 45] or deformable convolution [46] for feature alignment and fusion temporal information. However, they introduce auxiliary modules that are difficult to binarize. Other methods directly employ 2D or 3D convolutions [14, 30] for implicit alignment, which has a limited receptive field to deal with large motions in videos. *ii*) Binary convolutions encounter degradation in representation capability compared to their full precision counterparts [20]. Because binarizing real-valued activations and weights to 1-bit leads to information loss of absolute value and distribution characteristics.

In this paper, we design a binary raw video enhancement model (BRVE) to address these issues. Specifically, we introduce an efficient spatial-temporal shift operation to fully exploit the temporal redundancy for video enhancement. We use a cyclic temporal shift to fuse the features of frames in a local window and perform a spatial shift on features to enlarge the receptive field and handle the misalignment. These shift operations do not introduce extra parameters and enable parallel processing of multiple frames using 2D convolutions. Besides, we propose a distribution-aware binary convolution that can reduce the performance gap between binary convolutions and full precision ones. It employs a distribution-aware channel attention to extract a real-valued scale factor from the input activation with negligible computation. The scale factor injects distribution information of full precision input into vanilla binary convolutions to reduce the quantization error. As shown in Figure 1, our binarized low-light video enhancement method can reduce the model size and computation while having a comparable performance with full precision ones. Our main contributions can be summarized as follows:

- We build a compact binarized model for low-light raw video enhancement, which can achieve satisfactory results with low memory and computation.
- We design an easy-to-binarize spatial-temporal shift operation to tackle the alignment of features and aggregate temporal information for low-light video enhancement.
- We propose a distribution-aware binary convolution to mitigate the information loss of real-valued activations caused by binarization.

## 2. Related Work

In this section, we review datasets and methods for low-light video enhancement and various applications of BNNs.

### 2.1. Low-light Video Enhancement

Compared to low-light image enhancement, enhancing low-light videos is more challenging because obtaining paired low-light and clean, noise-free normal exposure videos is difficult. Therefore, some work uses camera noise model [36, 38, 39, 48, 54] or Generative Adversarial Networks (GAN) [31] to synthesize low-light noisy videos for training neural networks. For real-captured training data, Chen *et al.* [5] collects a static Dark Raw Video (DRV) dataset. To obtain paired low/normal-light videos with motion, some work adopts mechatronic systems [8, 35] to reproduce the same motion twice. Jiang *et al.* [14] builds a dual-camera system to shoot video pairs at once. A beam splitter is used to generate two identical light beams and one of them is weakened through a neutral density (ND) filter. To get rid of extra equipment, Fu *et al.* [10] shoots a sequence of dark/bright frame pairs on a 4K monitor to form a video pair. Among these datasets, we mainly focus on enhancing low-light raw videos. Because raw data is directly obtained from the sensor and is not processed by non-linear operations in the image signal processor (ISP). We use the linearly scaling then denoising pipeline for low-light raw video enhancement following previous methods [4, 5, 14].

Directly using low-light image enhancement methods without considering temporal consistency may cause a flickering problem. Besides, processing low-light video frame-wisely leads to inferior denoising performance because the temporal redundancy is not fully exploited. Recently, many deep-learning models have been developed to improve temporal consistency and make use of spatial-temporal information for low-light video enhancement and denoising [10, 14, 37, 47]. SMOID [14] exploits 3D convolutions to aggregate temporal features for video enhancement. Some methods like DVDNet [29] and TOFlow [45] use optical flow to align neighbor frames or features for denoising. RViDeNet [46] performs alignment with pyramidal deformable convolution for raw video denoising. ShiftNet [18] adopts grouped spatial-temporal shift to simplify feature aggregation. In order to improve the efficiency of the networks, FastDVD [30] uses two-stage cascaded U-Nets for implicit motion compensation. EMVD [23] leverages a fusion stage to efficiently reduce noise with former frames. However, these methods only focus on lightweight network design without exploring hardware-friendly approaches like model quantization and binarization.

### 2.2. Binary Neural Networks

Among the various deep neural network compression techniques, binary neural network (BNN) is an extreme form of network quantization. In a BNN, the network's weight parameters and activation values are represented using 1-bit values (*i.e.*-1 and +1) which can significantly reduce the storage and computational requirements [24]. The pioneer-

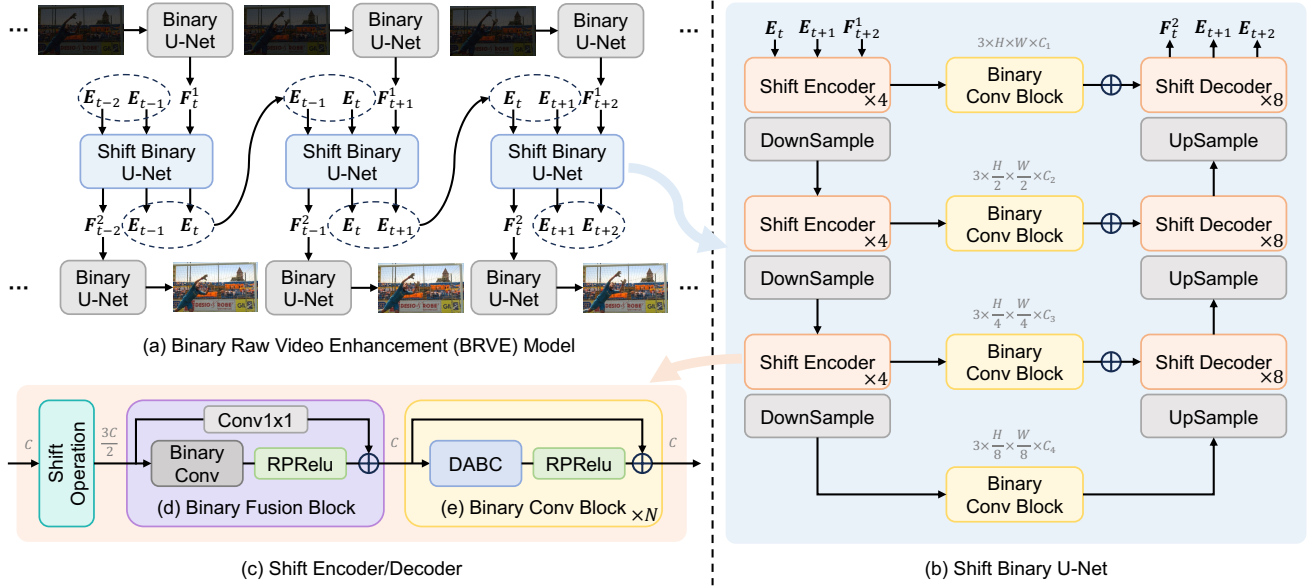


Figure 2. Overall architecture of BRVE model. (a) BRVE uses a shift binary U-Net for local feature fusion and exploits recurrent embeddings for long-range feature propagation. (b) Shift binary U-Net. (c) Shift encoder/decoder consists of a shift operation, a binary fusion block, and several binary conv blocks using the distribution-aware binary convolution (DABC).

ing work [7, 13] uses the sign function to obtain binarized activations and weights and optimize network parameters with the straight-through-estimator (STE). However, BNNs face the loss of precision due to quantization error, gradient error, and limited representation ability. To reduce the quantization error, XNOR-Net [26] utilizes scaling factors for weights and activations. IRNet [25] proposes to normalize the weights to retain information and increase information entropy. To reduce the gradient error, many efforts have been made to optimize the approximation of the sign function. Liu *et al.* [20] uses a piecewise quadratic function. Qin *et al.* [25] introduces an error decay estimator (EDE) with the tanh function. Xu *et al.* [43] proposes to decompose the sign function with Fourier series. Wu *et al.* [40] develops a Rectified STE (ReSTE) to balance the gradient estimating error and the training stability. To improve the capability of BNNs, Bi-real [20] preserves real-weight activations through a simple shortcut. ReActNet [21] and INSTA-BNN [17] learn a channel-wise threshold to improve the binary activation function for activations.

In addition to its widespread use in high-level vision tasks, BNNs are also applied in low-level vision [2, 15, 41, 42]. Jiang *et al.* [15] removes batch normalization (BN) layers and introduces a binary training mechanism for image super-resolution. Xia *et al.* [41] designs a basic binary convolution unit (BBCU) for image restoration. Cai *et al.* [2] proposes a BiSRNet for hyperspectral image reconstruction. Nevertheless, how to use binary networks to extract temporal information in videos has not been discussed.

### 3. Method

In this section, we first introduce our low-light raw video enhancement pipeline. Then we present the architecture of our binary raw video enhancement (BRVE) model. Finally, we elaborate on the details of the distribution-aware binary convolution and spatial-temporal shift operation.

#### 3.1. Binary Raw Video Enhancement Model

**Problem Formulation.** Given a consecutive sequence of low-light noisy raw video frames in Bayer pattern, denoted as  $\{I_1^B, I_2^B, \dots, I_T^B\}$ , where  $I_i^B \in \mathbb{R}^{H \times W}$  and  $T$  is the number of input frames. Following previous work [4, 46], we first pack each  $2 \times 2$  Bayer pattern into four color channels. Benefiting from the linear relationship of raw data with exposure levels, we utilize a predefined scaling factor  $r$  to amplify low-light videos to a proper brightness. These packed and amplified frames are denoted as  $\{I_1^P, I_2^P, \dots, I_T^P\}$ , where  $I_i^P \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 4}$ , which are then fed into the BRVE model to generate clean bright frames  $\{O_1^P, O_2^P, \dots, O_T^P\}$ .

**Overall Architecture.** The overall architecture of our BRVE model is illustrated in Figure 2 (a). In the first stage, we use a binary U-Net [27] to extract features for each input raw frame. The binary U-Net uses binary conv blocks as encoders and decoders. The structure of binary conv blocks is shown in Figure 2 (e). It adopts the distribution-aware binary convolution (DABC) to improve the capability of BNNs, which will be introduced in Section 3.2. As the high

system gain in low-light photography and the amplification factor increase the noise, the first U-Net can also serve as a pre-denoising step. As shown in Figure 2 (b), the main component of BRVE is a shift U-Net, which exploit shift encoders/decoders in each level. As illustrated in Figure 2 (c), it applies shift operation for spatial-temporal feature fusion in a local sliding window, which will be discussed in Section 3.3. In the last stage, we also use a binary U-Net to aggregate features in the former two stages for raw video reconstruction. Following previous BNN work [20, 41], we do not binarize the first convolution in the first U-Net to retain more information of raw input and the last convolution in the last U-Net to better maps features to bright and denoised raw frames.

**Recurrent Embedding.** As shown in Figure 2 (a), we add recurrent embeddings between neighbor windows for long-range feature propagation. For a local temporal window  $\{t-1, t, t+1\}$ , the shift binary U-Net takes two embedding features  $E_{t-1}, E_t$  from previous window and  $F_{t+1}^1$  as input. The first output  $F_{t-1}^1$  is then fed into the next stage and the latter two outputs  $E_t, E_{t+1}$  are used as recurrent embeddings for next window.

### 3.2. Distribution-Aware Binary Convolution

The main reason for the performance degradation of binary convolutions is the loss of full precision information in the convolution kernel weights and activations. To address this issue, real-valued scale factors [26, 44] are often adopted to multiply with the output of binary convolution. However, most of them are fixed during inference making it less flexible to use the information of different real-valued input activations. To fully exploit full precision information, we propose the distribution-aware binary convolution (DABC) that uses distribution-aware channel attention (DACA) to efficiently generate dynamic scale factors in terms of the distribution characteristic of input activations.

**Binary Convolution.** The core of the binary convolution is to binarize full precision activation  $A^f \in \mathbb{R}^{H \times W \times C}$  and kernel weight  $W^f \in \mathbb{R}^{C_{out} \times C \times K \times K}$  to a binary set  $\mathbb{B} = \{-1, +1\}$ . As shown in Figure 3 (a), the binarize procedure of the weight can be represented as

$$\begin{aligned} W^b &= \text{Sign}(W^f) = \begin{cases} +1, & W^f > 0 \\ -1, & W^f \leq 0 \end{cases}, \\ S_i &= \frac{\|W_i^f\|_1}{C \times K \times K}, \quad i = 1 \dots C_{out}, \end{aligned} \quad (1)$$

where  $W^b \in \mathbb{B}^{C_{out} \times C \times K \times K}$  is the binary kernel weight. Following previous work [26], the scale factor  $S \in \mathbb{R}^{C_{out}}$  is used for reducing the quantization error of the binary kernel weight. The real-valued activation is binarized by the RSign [21] function, *i.e.*,

$$A^b = \text{RSign}(A^f) = \text{Sign}(A'), \quad (2)$$

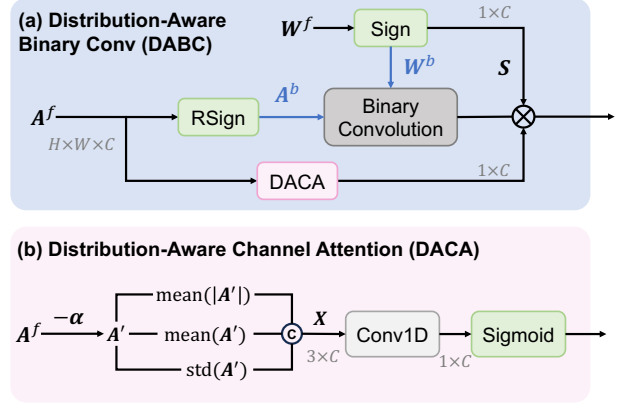


Figure 3. Distribution-Aware Binary Convolution (DABC).

where  $A^b \in \mathbb{B}^{H \times W \times C}$  is the binarized activation and  $A' = A^f - \alpha$ , where  $\alpha \in \mathbb{R}^C$  is a learnable channel-wise parameter that controls the threshold for binarizing the input activation. With binary activation and weight, the binary convolution  $\otimes$  can be efficiently computed using bit-wise operations

$$A^b \otimes W^b = \text{bitcount}(\text{XNOR}(A^b, W^b)). \quad (3)$$

**Distribution-Aware Channel Attention.** To reduce the information loss from real-valued activation, we first collect several channel-wise statistics from  $A'$ , which describes the distribution characteristic of the activation [17]. Inspired by the efficient channel attention module [34], we use a 1D convolution followed by a Sigmoid function to extract a distribution-aware scale factor from these statistics with negligible computation and parameters. As shown in Figure 3 (b), the DACA module can be represented as

$$\begin{aligned} X &= \text{Concat}(\text{mean}(|A'|), \text{mean}(A'), \text{std}(A')), \\ \text{DACA}(A') &= \text{Sigmoid}(\text{Conv1D}(X)), \end{aligned} \quad (4)$$

where  $X \in \mathbb{R}^{C \times 3}$  and  $\text{DACA}(A') \in \mathbb{R}^{C \times 1}$ . Finally, the distribution-aware binary convolution is defined as

$$Y = (A^b \otimes W^b) \odot S \odot \text{DACA}(A') \quad (5)$$

where  $Y \in \mathbb{Z}^{H \times W \times C_{out}}$  is the output activation rescaled with the factors from weights and DACA module.

**Basic Binarized Modules.** As shown in Figure 2 (e), a binary convolution and a RPRelu [21] function form the binary conv block. RPRelu function can be formulated as

$$\text{RPRelu}(Y) = \begin{cases} Y_i - \gamma_i + \zeta_i, & Y_i > \zeta_i \\ \beta_i(Y_i - \gamma_i) + \zeta_i, & Y_i \leq \zeta_i, \end{cases} \quad (6)$$

where  $\gamma_i, \zeta_i, \beta_i \in \mathbb{R}^{C_{out}}$  are learnable parameters. As incorporating a full precision shortcut connection is crucial in

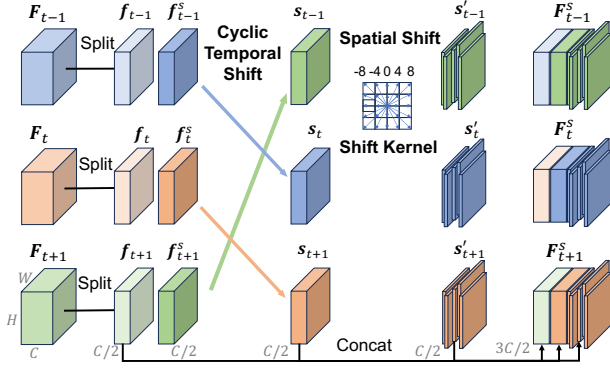


Figure 4. Spatial-temporal shift operation.

binary convolution to compensate for the information loss, we add the full precision input  $A^f$  with the output of binary convolution following previous work [20, 41]. However, binary convolution needs to change the channel number of activation (*i.e.*  $C \neq C_{out}$ ) in the shift encoder/decoder, which disables the addition of shortcut as well as the multiplication of DACA module. As shown in Figure 2 (d), we use a full precision  $1 \times 1$  convolution to replace them to match the channel in the binary fusion block. To minimize computation cost, the binary fusion block is only used once in each shift encoder/decoder. We use bilinear upsampling and average pooling for downsampling in the binary U-Net.

### 3.3. Spatial-Temporal Shift Operation

Spatial-temporal self-similarity is a crucial property for video denoising. Previous methods either use complicated models that are difficult to quantize [19, 45, 46] or directly use convolution [30] for temporal feature fusion. Inspired by the ShiftNet [18], we introduce a temporal-spatial shift operation, which propagates information between neighbor frames through the temporal shift and enlarges the receptive field of the network with the spatial shift. The shift operation is performed at the beginning of each shift encoder/decoder as shown in Figure 2 (c). It is efficient and parameter-free because it only transforms the input features and uses subsequent binarized modules described in Section 3.2 for feature fusion.

**Cyclic Temporal Shift.** In order to enable bi-directional feature propagation, ShiftNet [18] performs forward and backward temporal shift alternatively in a sequence of features. However, caching these features requires a significant storage overhead and prevents online processing [19]. We propose a cyclic temporal shift to aggregate features in a three-frame local window to reduce the feature cache.

As shown in Figure 4, the inputs of shift operation are three consecutive features  $\{F_{t-1}, F_t, F_{t+1}\}$ . Each of them is first split into two parts along the channel dimension

$$[f_i, f_i^s] = \text{Split}(F_i), i \in \{t-1, t, t+1\}, \quad (7)$$

where  $F_i \in \mathbb{R}^{H \times W \times C}$  and  $f_i, f_i^s \in \mathbb{R}^{H \times W \times C/2}$ . The first part  $f_i$  keeps information of frame  $i$  and the second part  $f_i^s$  is passed to the neighbor frame to perform spatial-temporal shift and obtain  $F_i^s$  for feature fusion.

In cyclic temporal shift, the second part features  $f_{t-1}^s, f_t^s$  shift to next position and feature  $f_{t+1}^s$  becomes the first one, as shown in Figure 4. After the temporal shift, we can obtain a reordered set of features

$$S^t = \{s_{t-1}, s_t, s_{t+1}\} = \{f_{t+1}^s, f_{t-1}^s, f_t^s\}. \quad (8)$$

However, performing the cyclic temporal shift operation once is insufficient to integrate each feature with the other two features in the local window. We also shift the features in the opposite direction along the temporal dimension and the shifted set  $S^t$  becomes  $\{f_t^s, f_{t+1}^s, f_{t-1}^s\}$ . In each level of shift binary U-Net, we conduct cyclic temporal shift for two directions alternatively to fuse the features adequately.

**Spatial Shift.** Both camera motion and object movement in the scene can result in the misalignment of features between frames. Nevertheless, directly fusing the first part feature  $f_i$  with shifted feature  $s_i$  by binary convolutions cannot handle large motions. Following previous work [18], we adopt the spatial shift operation to enlarge the receptive field of subsequent binary convolution blocks and cope with misalignment in all possible directions.

Specifically, we first define a spatial shift kernel

$$K = \{(x, y) | x, y \in \{-8, -4, 0, 4, 8\}, (x, y) \neq (0, 0)\}, \quad (9)$$

where  $|K| = 24$  and each element  $(x, y)$  in the kernel represents the number of pixels to shift along the corresponding axis. We divide each temporally shifted feature  $s_i$  into  $|K|$  slices. Each slice spatially shifts in a different direction according to the shift kernel

$$s'_{i,j} = \text{Shift}(s_{i,j}, (x_j, y_j)), j = 1 \dots |K|, \quad (10)$$

where  $s_{i,j}$  is the  $j$ -th slice of  $s_i$  and empty area of  $s'_{i,j}$  caused by spatial shift is filled by zero. Then we concatenate all shifted slices to obtain three spatially shifted features  $\{s'_{i-1}, s'_i, s'_{i+1}\}$ , where  $s'_i = \text{Concat}(s'_{i,1}, \dots, s'_{i,|K|})$ . Finally, we concatenate spatial-temporal shift features together with the unshifted feature for each frame

$$F_i^s = \text{Concat}(f_i, s'_i, s_i), i \in \{t-1, t, t+1\}, \quad (11)$$

where  $F_i^s \in \mathbb{R}^{H \times W \times 3C/2}$  is then fed into a binary fusion block and several binary conv blocks for spatial-temporal feature fusion, as shown in Figure 2 (c).

## 4. Experiments

In this section, we evaluate our BRVE model on two low-light raw video enhancement datasets. We also conduct extensive experiments to analyze our proposed model.

Method	Gain0			Gain15			Gain30			Params (M)	FLOPs (G)
	PSNR $\uparrow$	SSIM $\uparrow$	ST-RRED $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	ST-RRED $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	ST-RRED $\downarrow$		
SMOID [14]	39.74	0.9733	0.0446	39.43	0.9751	0.0461	40.06	0.9769	0.0431	12.21	15.11
RViDeNet [46]	41.09	0.9778	0.0510	40.94	0.9803	0.0494	41.60	0.9818	0.0500	8.57	65.77
FastDVD [30]	40.37	0.9724	0.0765	40.66	0.9768	0.0664	40.59	0.9763	0.1812	2.48	10.50
EMVD-L [23]	41.10	0.9785	0.0657	40.34	0.9804	0.0691	41.13	0.9819	0.0796	9.6	17.52
EMVD-S [23]	39.71	0.9707	0.0732	39.23	0.9735	0.0804	39.88	0.9756	0.0875	0.81	1.66
LLRVD [10]	41.51	0.9799	0.0388	41.75	0.9823	0.0330	42.13	0.9840	0.0350	6.29	46.14
FloRNN [19]	41.39	0.9801	0.0468	40.74	0.9823	0.0560	41.55	0.9842	0.0558	10.49	24.57
ShiftNet [18]	<b>42.11</b>	<b>0.9836</b>	<b>0.0328</b>	<b>42.28</b>	<b>0.9848</b>	<b>0.0273</b>	<b>42.70</b>	<b>0.9863</b>	<b>0.0280</b>	13.38	32.87
BNN [13]	33.44	0.8417	0.2242	33.16	0.8524	0.2961	34.25	0.8508	0.2511	0.29	1.38
Bireal [20]	36.81	0.9227	0.1320	36.71	0.9424	0.1572	37.34	0.9413	0.1142	0.29	1.38
IRNet [25]	33.29	0.8119	0.1848	33.52	0.8378	0.1834	34.35	0.8543	0.1758	0.29	1.38
ReActNet [21]	37.20	0.9245	0.1502	37.27	0.9514	0.1484	37.93	0.9449	0.1088	0.31	1.55
BTM [15]	37.87	0.9445	0.0908	38.14	0.9545	0.1081	38.14	0.9581	0.1058	0.28	1.35
BBCU [41]	39.92	0.9736	0.0745	40.11	0.9756	0.0660	40.48	0.9780	0.0637	0.3	1.47
BRVE (Ours)	<b>40.05</b>	<b>0.9742</b>	<b>0.0639</b>	<b>40.25</b>	<b>0.9765</b>	<b>0.0557</b>	<b>40.64</b>	<b>0.9786</b>	<b>0.0570</b>	0.3	1.49

Table 1. Quantitative results on SMOID dataset. Our BRVE model outperforms other BNN methods at all three gain levels. BRVE also surpasses the performance of full precision EMVD-S with similar computational costs.

## 4.1. Experiment Settings

**Datasets.** Two low-light raw video datasets are adopted in our experiment. SMOID [14] has 309 video pairs of 103 scenes. The resolution of each frame is  $480 \times 640$ . Three low-light videos are captured in each scene using different ADC gain levels (*i.e.* 0, 15, and 30). We use 70 scenes for training, 16 for validation, and 17 for testing. LLRVD [10] contains 210 video pairs of 70 scenes. The resolution of each frame is  $1400 \times 2600$ . For each scene, three videos with different low-light ratios are captured by fixing ISO and adjusting exposure time. We use 60 scenes for training, 4 for validation, and 6 for testing. These two datasets are captured with different cameras and SMOID has different noise distribution at different gain levels, which can comprehensively evaluate our BRVE model.

**Training Details.** We use a sequence of 10 frames to train the models on both datasets. The batch size is set to 1 and the input raw patch size is set to  $256 \times 256$ . We adopt Charbonnier loss [3] to train 100K iterations using Adam optimizer [16] and the cosine annealing scheduler with the learning rate set to  $2 \times 10^{-4}$ . We implement BRVE with PyTorch and train it on an NVIDIA RTX 3090 GPU.

**Metrics.** We use the average peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) of all enhanced raw frames to assess the restoration quality of each method. In order to take temporal distortion into account, we use the spatio-temporal reduced reference entropic differences (ST-RRED) [28] to evaluate the video quality. Higher PSNR and SSIM mean better restoration quality and lower ST-RRED indicates better video fidelity.

Method	PSNR $\uparrow$	SSIM $\uparrow$	ST-RRED $\downarrow$
SMOID [14]	37.07	0.9580	0.0391
RViDeNet [46]	37.39	0.9620	0.0400
FastDVD [30]	36.39	0.9450	0.0724
EMVD-L [23]	37.00	0.9576	0.0473
EMVD-S [23]	36.70	0.9527	0.0527
LLRVD [10]	37.74	0.9650	0.0347
FloRNN [19]	37.47	0.9634	0.0377
ShiftNet [18]	<b>37.87</b>	<b>0.9661</b>	<b>0.0346</b>
BNN [13]	31.04	0.7393	0.0817
Bireal [20]	35.90	0.9356	0.0701
IRNet [25]	34.10	0.8768	0.0967
ReActNet [21]	35.78	0.9330	0.0697
BTM [15]	36.56	0.9499	0.0556
BBCU [41]	36.95	0.9575	0.0457
BRVE (ours)	<b>37.07</b>	<b>0.9581</b>	<b>0.0455</b>

Table 2. Quantitative results on LLRVD dataset.

**Efficiency Evaluation.** Following previous BNN work [2, 20, 26, 41], we calculate BNN operations by  $OPs^b = OPs^f/64$  where  $OPs^f$  is full precision operations, and the parameters of BNNs is computed by  $Params^b = Params^f/32$  where  $Params^f$  is real-valued parameters. The overall FLOPs is  $OPs^b + OPs^f$ , we use each model to process a video with 100 frames and a resolution of  $256 \times 256$  for counting the per-frame FLOPs. The total number of parameters is  $Params^b + Params^f$ .

## 4.2. Compare with State-of-the-arts

**Comparison Methods.** We compare our method with various full precision video denoising networks including

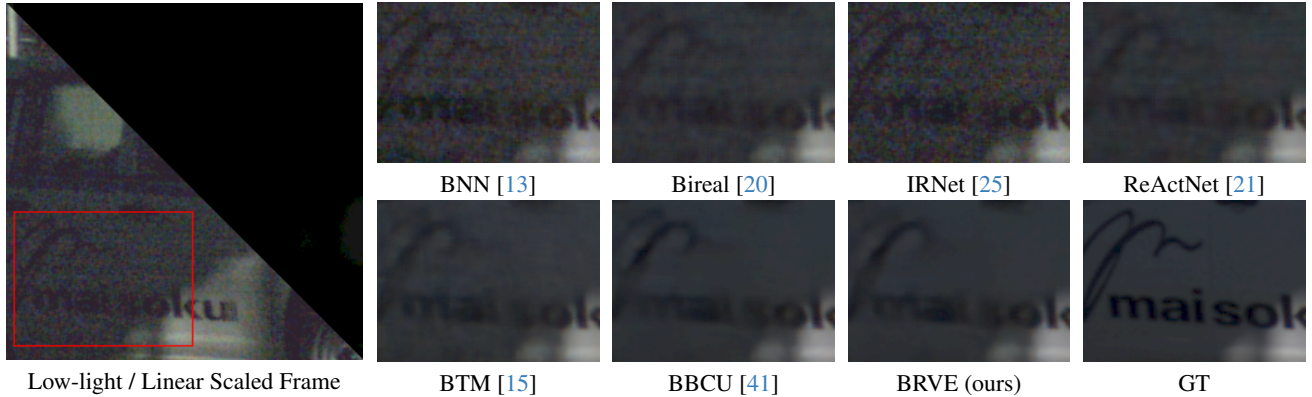


Figure 5. Visual comparison of different low-light video enhancement methods on SMOID datasets.

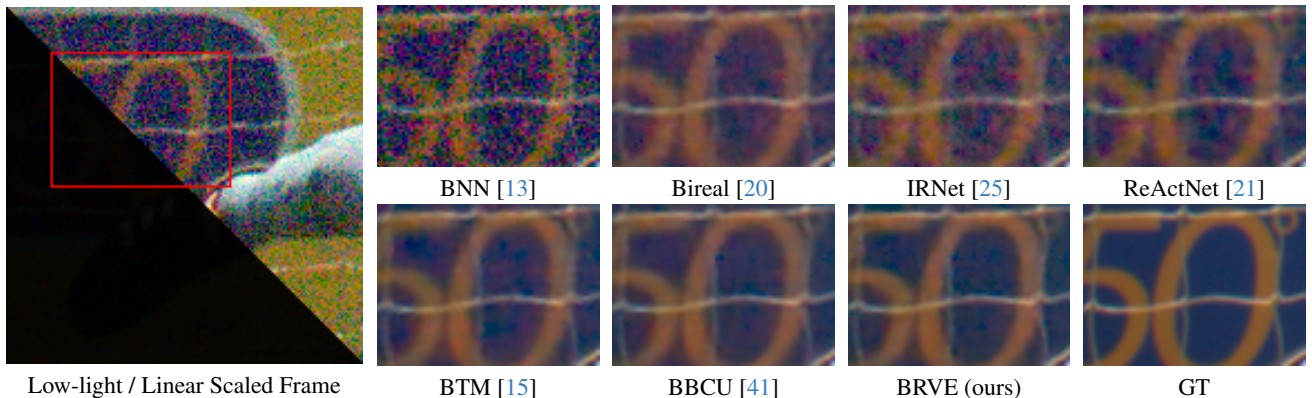


Figure 6. Visual comparison of different low-light video enhancement methods on LLRVD datasets.

SMOID [14], RViDeNet [46], FastDVD [30], EMVD [23], LLRVD [10], FloRNN [19], and ShiftNet [18]. We train two versions of the lightweight EMVD model, denoted as EMVD-L (large) and EMVD-S (small). We also compare our distribution-aware binary convolution with other binarization methods, including BNN [13], Bireal [20], IRNet [25], ReActNet [21], BTM [15], and BBCU [41].

**Quantitative Comparison.** We illustrate the performance on the SMOID dataset and the model complexity in Table 1. The upper part shows the results of full precision models, while the lower part shows the results of the BNNs. It can be observed that ShiftNet [18] outperforms other full precision methods, which shows the effectiveness of the shift operation. Besides, we can find that all the BNN methods greatly reduce the parameters and operations. Compared with the state-of-the-art BNN module BBCU [41], our distribution-aware binary convolution improves the performance of BNN at all gain levels with only 1.4% additional computation cost.

Table 2 demonstrates the quantitative results of the LLRVD dataset. Our method not only surpasses the lightweight full precision EMVD-S model with similar FLOPs by 0.49 dB but also achieves comparable results to the EMVD-L model while utilizing only 9.4% of its FLOPs

mean(  ·  )	mean(·) std(·)	PSNR ↑	SSIM ↑	ST-RRED ↓
✗	✗	36.95	0.9575	0.0457
✗	✓	37.02	0.9578	0.0457
✓	✗	37.01	0.9577	<b>0.0450</b>
w/o shift		36.70	0.9544	0.0517
BRVE (ours)		<b>37.07</b>	<b>0.9581</b>	0.0455

Table 3. Ablation study on the DACA module and shift operation.

and 3.3% of the parameters.

**Visual Comparison.** The qualitative results on the SMOID dataset and LLRVD dataset are illustrated in Figures 5 and 6, respectively. We show both low-light and noisy linear scaled frame in the left. Benefiting from the distribution-aware binary convolution, our BRVE exhibits superior visual quality, effectively removing the noise and restoring more details compared to other BNN methods.

### 4.3. Analysis of the Proposed Method

**Ablation Study.** We first conduct the ablation study to verify the effect of different channel-wise real-valued statistics on our DACA module. The result is shown in Table 3, where the first column represents whether to use the ab-

Method	PSNR $\uparrow$	SSIM $\uparrow$	ST-RRED $\downarrow$
Raw2Raw+ISP	<b>30.46</b>	<b>0.8399</b>	<b>0.2021</b>
Raw2RGB	27.42	0.8131	0.2952
RGB2RGB	24.84	0.7988	0.5951
Raw2Raw+ISP+H264	<b>30.19</b>	<b>0.8416</b>	<b>0.1828</b>
RGB2RGB+H264	24.75	0.8158	0.5622
H264+RGB2RGB	20.03	0.6130	1.4228

Table 4. Comparison of different low-light video enhancement settings on LLRVD dataset. Metrics are computed in RGB domain.

solute value, the second column represents whether to use the mean and standard deviation. We can observe that the performance can be improved after injecting each of these statistics into binary convolution. BRVE model obtains a better result by using all of these statistics to describe the distribution characteristic of full precision activations.

We also perform an ablation study on the spatial-temporal shift operation. As shown in the fourth row of Table 3, the performance declines significantly without the shift operation, which indicates the effectiveness of shift operation in exploiting spatial-temporal self-similarity for low-light raw video denoising.

**Comparison on Video Enhancement Settings.** Table 4 compares different low-light video enhancement settings, including raw-to-raw (Raw2Raw), raw-to-RGB (Raw2RGB), and RGB-to-RGB (RGB2RGB) on the LLRVD dataset. The raw output of the Raw2Raw setting is converted to RGB domain with ISP. All metrics are evaluated in RGB domain with the ground truth (GT).

The Raw2Raw setting adopted in this paper shows the best performance, where the low-light enhancement model serves as a pre-processing step before ISP. In the Raw2RGB setting, the network jointly learns to enhance low-light videos and the non-linear process of ISP. It is less effective than the Raw2Raw setting because it is difficult for the network to learn the ISP accurately. In the RGB2RGB setting, we use the network to enhance dark RGB videos, which leads to a significant performance decline. The reason is that RGB videos are quantized into 8-bit and the information of low-intensity pixels in the dark is lost.

In addition, we also discuss the influence of video compression on low-light video enhancement. As shown in Table 4, H264 compression has little impact on the quality of enhanced videos. However, enhancing compressed low-light videos causes severe degradation, which demonstrates the importance of enhancing the video in the early stages of the image processing pipeline on edge devices.

**Effect of Sliding Stride.** We discuss the influence of the sliding window stride on the recurrent propagation. In the BRVE model, the stride is set to one and the shift binary U-Net outputs one feature at a time. The last two features are used as recurrent embeddings. We can also set the stride to

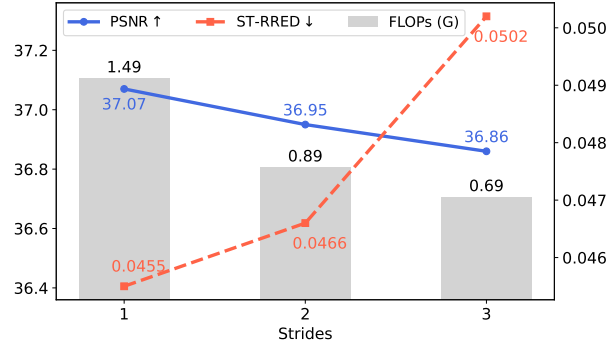


Figure 7. Effect of the sliding window stride on recurrent propagation. A larger sliding stride can further boost the efficiency while compromising the temporal consistency.

two or three to further improve the model efficiency. When setting the stride to two, the shift binary U-Net outputs two features at a time and uses the last feature as the recurrent embedding. When setting the stride to three, the shift binary U-Net outputs all three features in the local window, and no recurrent embedding is added between windows.

As shown in Figure 7, using a stride of two reduces about 40% FLOPs compared to the original BRVE model. But the performance drops 0.12 dB in PSNR, which indicates that using more features in recurrent embedding is helpful for video enhancement as more long-term history information is incorporated. Using a stride of three to process each window independently can maximize efficiency. But the large increment of ST-RRED shows that the temporal consistency is greatly impaired without recurrent propagation.

## 5. Conclusion

In this paper, we propose a BRVE model for low-light raw video enhancement. First, we propose the distribution-aware convolution to mitigate the performance gap between BNNs and CNNs. It extracts full precision distribution characteristic information with DACA module to improve the capability of binary convolutions. Second, we introduce a spatial-temporal shift operation to fuse temporal information and maintain temporal consistency. It is parameter-free and enables efficient parallel processing for features in a local window. Experiments on two low-light raw video datasets demonstrate our BRVE model can achieve comparable performance with some full precision models using less memory and computational costs.

**Acknowledgement** This work was supported by the National Natural Science Foundation of China (62331006, 62171038, 62088101, and 62271414), the R&D Program of Beijing Municipal Education Commission (KZ202211417048), Science Fund for Distinguished Young Scholars of Zhejiang Province (LR23F010001), and the Fundamental Research Funds for the Central Universities.



## References

- [1] Pablo Arias and Jean-Michel Morel. Video denoising via empirical bayesian estimation of space-time patches. *J. Math. Imaging Vis.*, 60(1):70–93, 2018. [2](#)
- [2] Yuanhao Cai, Yuxin Zheng, Jing Lin, Haoqian Wang, Xin Yuan, and Yulun Zhang. Binarized spectral compressive imaging. *arXiv preprint arXiv:2305.10299*, 2023. [3](#), [6](#)
- [3] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, pages 168–172, 1994. [6](#)
- [4] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, pages 3291–3300, 2018. [1](#), [2](#), [3](#)
- [5] Chen Chen, Qifeng Chen, Minh N. Do, and Vladlen Koltun. Seeing motion in the dark. In *ICCV*, pages 3184–3193, 2019. [1](#), [2](#)
- [6] Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *IJCV*, 131(8): 2198–2218, 2023. [1](#)
- [7] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NeurIPS*, pages 3123–3131, 2015. [3](#)
- [8] Huiyuan Fu, Wenkai Zheng, Xicong Wang, Jiaxuan Wang, Heng Zhang, and Huadong Ma. Dancing in the dark: A benchmark towards general low-light video enhancement. In *ICCV*, pages 12877–12886, 2023. [2](#)
- [9] Ying Fu, Yang Hong, Linwei Chen, and Shaodi You. LE-GAN: Unsupervised low-light image enhancement network using attention module and identity invariant loss. *Knowledge-Based Systems*, 240:108010, 2022. [1](#)
- [10] Ying Fu, Zichun Wang, Tao Zhang, and Jun Zhang. Low-light raw video denoising with a high-quality realistic motion dataset. *IEEE TMM*, pages 1–13, 2022. [1](#), [2](#), [6](#), [7](#)
- [11] Yang Hong, Kaixuan Wei, Linwei Chen, and Ying Fu. Crafting object detection in very low light. In *BMVC*, 2021. [1](#)
- [12] Haofeng Huang, Wenhan Yang, Yueyu Hu, Jiaying Liu, and Ling-Yu Duan. Towards low light enhancement with RAW images. *IEEE TIP*, 31:1391–1405, 2022. [1](#)
- [13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NeurIPS*, pages 4107–4115, 2016. [1](#), [3](#), [6](#), [7](#)
- [14] Haiyang Jiang and Yinqiang Zheng. Learning to see moving objects in the dark. In *ICCV*, pages 7323–7332, 2019. [1](#), [2](#), [6](#), [7](#)
- [15] Xinrui Jiang, Nannan Wang, Jingwei Xin, Keyu Li, Xi Yang, and Xinbo Gao. Training binary neural network without batch normalization for image super-resolution. In *AAAI*, pages 1700–1707, 2021. [3](#), [6](#), [7](#)
- [16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [6](#)
- [17] Changhun Lee, Hyungjun Kim, Eunhyeok Park, and Jae-Joon Kim. INSTA-BNN: Binary neural network with instance-aware threshold. In *ICCV*, pages 17325–17334, 2023. [3](#), [4](#)
- [18] Dasong Li, Xiaoyu Shi, Yi Zhang, Ka Chun Cheung, Simon See, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. A simple baseline for video restoration with grouped spatial-temporal shift. In *CVPR*, pages 9822–9832, 2023. [1](#), [2](#), [5](#), [6](#), [7](#)
- [19] Junyi Li, Xiaohe Wu, Zhenxing Niu, and Wangmeng Zuo. Unidirectional video denoising by mimicking backward recurrent modules with look-ahead forward ones. In *ECCV*, pages 592–609, 2022. [1](#), [2](#), [5](#), [6](#), [7](#)
- [20] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, pages 747–763, 2018. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [21] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. ReActNet: Towards precise binary neural network with generalized activation functions. In *ECCV*, pages 143–159, 2020. [3](#), [4](#), [6](#), [7](#)
- [22] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen O. Egiazarian. Video denoising using separable 4d nonlocal spatiotemporal transforms. In *Image Processing: Algorithms and Systems IX*, pages 9–19, 2011. [2](#)
- [23] Matteo Maggioni, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *CVPR*, pages 3466–3475, 2021. [2](#), [6](#), [7](#)
- [24] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *PR*, 105:107281, 2020. [1](#), [2](#)
- [25] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *CVPR*, pages 2247–2256, 2020. [3](#), [6](#), [7](#)
- [26] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542, 2016. [3](#), [4](#), [6](#)
- [27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. [3](#)
- [28] Rajiv Soundararajan and Alan C. Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE TCSVT*, 23(4):684–694, 2013. [6](#)
- [29] Matias Tassano, Julie Delon, and Thomas Veit. DVDNET: A fast network for deep video denoising. In *ICIP*, pages 1805–1809, 2019. [2](#)
- [30] Matias Tassano, Julie Delon, and Thomas Veit. FastDVDnet: Towards real-time deep video denoising without flow estimation. In *CVPR*, pages 1351–1360, 2020. [2](#), [5](#), [6](#), [7](#)
- [31] Danai Triantafyllidou, Sean Moran, Steven McDonagh, Sarah Parisot, and Gregory G. Slabaugh. Low light video enhancement using synthetic data produced with an intermediate domain mapping. In *ECCV*, pages 103–119, 2020. [2](#)
- [32] Huan Wang and Yun Fu. Trainability preserving neural structured pruning. In *ICLR*, 2023. [1](#)
- [33] Huan Wang, Suhas Lohit, Michael N. Jones, and Yun Fu. What makes a "good" data augmentation in knowledge distillation - A statistical perspective. In *NeurIPS*, 2022. [1](#)

- [34] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *CVPR*, pages 11531–11539, 2020. 4
- [35] Ruixing Wang, Xiaogang Xu, Chi-Wing Fu, Jiangbo Lu, Bei Yu, and Jiaya Jia. Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment. In *ICCV*, pages 9680–9689, 2019. 2
- [36] Wei Wang, Xin Chen, Cheng Yang, Xiang Li, Xuemei Hu, and Tao Yue. Enhancing low light videos by exploring high sensitivity camera noise. In *ICCV*, pages 4110–4118, 2019. 2
- [37] Zichun Wang, Yulun Zhang, Debing Zhang, and Ying Fu. Recurrent self-supervised video denoising with denser receptive field. In *ACM MM*, pages 7363–7372, 2023. 2
- [38] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *CVPR*, pages 2755–2764, 2020. 2
- [39] Kaixuan Wei, Ying Fu, Yinqiang Zheng, and Jiaolong Yang. Physics-based noise modeling for extreme low-light photography. *IEEE TPAMI*, 44(11):8520–8537, 2022. 2
- [40] Xiao-Ming Wu, Dian Zheng, Zuhao Liu, and Wei-Shi Zheng. Estimator meets equilibrium perspective: A rectified straight through estimator for binary neural networks training. In *ICCV*, pages 17055–17064, 2023. 3
- [41] Bin Xia, Yulun Zhang, Yitong Wang, Yapeng Tian, Wenming Yang, Radu Timofte, and Luc Van Gool. Basic binary convolution unit for binarized image restoration network. In *ICLR*, 2023. 3, 4, 5, 6, 7
- [42] Jingwei Xin, Nannan Wang, Xinrui Jiang, Jie Li, Heng Huang, and Xinbo Gao. Binarized neural network for single image super resolution. In *ECCV*, pages 91–107, 2020. 3
- [43] Yixing Xu, Kai Han, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Learning frequency domain approximation for binary neural networks. In *NeurIPS*, pages 25553–25565, 2021. 3
- [44] Zhe Xu and Ray C. C. Cheung. Accurate and compact convolutional neural networks with trained binarization. In *BMVC*, page 19, 2019. 4
- [45] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 2, 5
- [46] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *CVPR*, pages 2298–2307, 2020. 2, 3, 5, 6, 7
- [47] Yuhang Zeng, Yunhao Zou, and Ying Fu.  $3D^2Unet$ : 3d deformable unet for low-light video enhancement. In *PRCV*, pages 66–77, 2021. 2
- [48] Fan Zhang, Yu Li, Shaodi You, and Ying Fu. Learning temporal consistency for low light video enhancement from single images. In *CVPR*, pages 4967–4976, 2021. 2
- [49] Fan Zhang, Shaodi You, Yu Li, and Ying Fu. Learning rain location prior for nighttime deraining. In *ICCV*, pages 13148–13157, 2023. 1
- [50] Tao Zhang, Ying Fu, and Cheng Li. Hyperspectral image denoising with realistic data. In *ICCV*, pages 2228–2237, 2021. 1
- [51] Tao Zhang, Ying Fu, and Jun Zhang. Guided hyperspectral image denoising with realistic data. *IJCV*, 130(11):2885–2901, 2022. 1
- [52] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 1
- [53] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In *ICLR*, 2022. 1
- [54] Yunhao Zou and Ying Fu. Estimating fine-grained noise model via contrastive learning. In *CVPR*, pages 12682–12691, 2022. 2
- [55] Yunhao Zou, Chenggang Yan, and Ying Fu. RawHDR: High dynamic range image reconstruction from a single raw image. In *ICCV*, pages 12334–12344, 2023. 1