Deep Parametric 3D Filters for Joint Video Denoising and Illumination Enhancement in Video Super Resolution

Xiaogang Xu[1,2], **Ruixing Wang** [2], **Chi-Wing Fu** [1], **Jiaya Jia** [1]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] SmartMore
{xgxu, cwfu, leojia}@cse.cuhk.edu.hk, ruixing.wang@smartmore.com

Abstract

Despite the quality improvement brought by the recent methods, video super-resolution (SR) is still very challenging, especially for videos that are low-light and noisy. The current best solution is to subsequently employ best models of video SR, denoising, and illumination enhancement, but doing so often lowers the image quality, due to the inconsistency between the models. This paper presents a new parametric representation called the *Deep Parametric 3D Filters* (DP3DF), which incorporates local spatiotemporal information to enable simultaneous denoising, illumination enhancement, and SR efficiently in a single encoder-and-decoder network. Also, a dynamic residual frame is jointly learned with the DP3DF via a shared backbone to further boost the SR quality. We performed extensive experiments, including a large-scale user study, to show our method's effectiveness. Our method *consistently* surpasses the best state-of-the-art methods on all the challenging real datasets with top PSNR and user ratings, yet having a very fast run time. The code is available at https://github.com/xiaogang00/DP3DF.

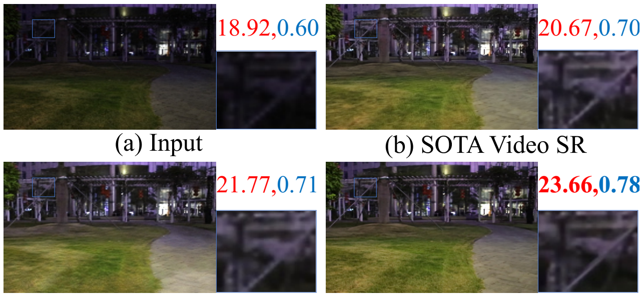
Introduction

The goal of video super resolution (SR) is to produce high-resolution videos from low-resolution video inputs. While promising results are demonstrated on general videos, existing approaches typically do not work well on videos that are low-light and noisy. Yet, such a setting is very common in practice, e.g., applying SR to enhance noisy videos taken in a dark and high-contrast environment.

Fundamentally, video denoising and video illumination enhancement are very different tasks from video SR: the former deals with noise and brightness in videos whereas the latter deals with the video resolution. Hence, to map a low-resolution, low-light, and noisy (LLN) video to a high-resolution, normal-light, and noise-free (HNN) video, the current best solution is to collectively use the best network model of each task by cascading models in a certain order.

However, doing so has several drawbacks. First, the network complexity is threefold, resulting in a slow inference, as we have to subsequently run three separate network models for denoising, illumination enhancement, and SR. Also, as the three networks are trained separately, we cannot ensure their consistency, e.g., artifacts from a preceding denoising or illumination enhancement network could be amplified by the subsequent SR network; see Fig. 1 (c). Alternatively, one may try to cascade and train all networks end-to-end. Yet, existing networks for video SR, denoising, and illumination enhancement often take multiple frames as input and output only a single frame, so a subsequent network cannot obtain sufficient inputs from the preceding one. Also, these networks are rather complex, so it is hard to fine-tune them together for high performance.

Another approach is to use parallel branches of different purposes in a framework. However, as the branches are separated from one another, their connections are weak for joint learning. Also, the input to all branches should be identical, while the sizes of their outputs are inconsistent: the output size of the SR branch is larger than its input size, while the other branches have same input and output sizes. Hence, how to achieve various purposes with one common branch and representation is worth to be considered. Further, we eventually will need to infer the different branches to produce the final results, which is time costly.



(a) Input (b) SOTA Video SR (c) Combined Strategy (d) Ours

Figure 1: An example frame (a) from a challenging under-exposed video enhanced by (b) a SOTA SR method (Xiang et al. 2020); (c) SOTA methods in video denoising (Tassano, Delon, and Veit 2020) + video illumination enhancement (Zhang et al. 2021) + video SR (Xiang et al. 2020); and (d) our approach. We can see that (d) is more sharp with distinct contrast, less noise, and less aliasing vs. (b) & (c). Please zoom to view the details. Score is PSNR/SSIM.

In this paper, we present a new solution to map LLN videos to HNN videos within a single end-to-end network. The core of our solution is the *Deep Parametric 3D Filter* (*DP3DF*), a novel dynamic-filter representation we formulated collectively for video SR, illumination enhancement, and denoising. This is the first work that we are aware of in exploring an efficient architecture for simultaneous video SR, denoising, and illumination enhancement. Beyond the existing works with dynamic filters, our DP3DF considers the burst from adjacent frames. Hence, DP3DF can effectively exploit local spatiotemporal neighboring information and complete the mapping from LLN video to HNN video in a single encoder-and-decoder network. Also, we show that general dynamic filters in existing works are just special cases of our DP3DF. Further, we set up an additional branch for learning dynamic residual frames on top of the core encoder-and-decoder network, so we can share the backbone for learning the DP3DF and residual frames to promote the overall performance.

To demonstrate the quality of our method, we conducted comprehensive experiments to compare our method with a rich set of state-of-the-art methods on two public video datasets SMID (Chen et al. 2019) and SDSD (Wang et al. 2021), which provide static and dynamic low- and normal-light video pairs. Through various quantitative and qualitative evaluations, including a large-scale user study with 80 participants, we show the effectiveness of our DP3DF framework over SOTA SR methods and also different combinations of SOTA video methods on illumination enhancement, denoising, and SR, both quantitatively and qualitatively. Our DP3DF framework surpasses the SOTA methods with top PSNR and user ratings consistently. In summary, our contributions are threefold:

- This is the first exploration of directly mapping LLN to HNN videos within a single-stage end-to-end network.
- This is the first work we are aware of that simultaneously achieves video SR, denoising, and illumination enhancement via our DP3DF representation.
- Extensive experiments are conducted on two real-world video datasets,demonstrating our superior performance.

Related Work

Video SR. Video SR aims to reconstruct a high-resolution frame from a low-resolution frame together with the associated adjacent frames. The key problem is on how to align the adjacent frames temporally with the center one. Several video SR methods (Caballero et al. 2017; Tao et al. 2017; Sajjadi, Vemulapalli, and Brown 2018; Wang et al. 2018; Xue et al. 2019) use optical flow for an explicit temporal alignment. However, it is hard to obtain accurate flow and the flow warping may introduce artifacts in the aligned frames. To leverage the temporal information, recurrent neural networks are adopted in some video SR methods (Huang, Wang, and Wang 2017; Lim and Lee 2017), e.g., the convolutional LSTMs (Shi et al. 2015). However, without an explicit temporal alignment, these RNN-based networks have limited capability in handling complex motions. Later, dynamic filters and deformable convolutions are exploited for temporal alignment. DUF (Jo et al. 2018) utilizes a dynamic filter to implement simple temporal alignment without motion estimation, whereas TDAN (Tian et al. 2020) and EDVR (Wang et al. 2019c) employ the deformable alignment in single- or multi-scale feature levels.

Video denoising. Early approaches are mostly patch-based, e.g., V-BM4D (Maggioni et al. 2012) and VNLB (Arias and Morel 2018), which extend from BM3D (Dabov et al. 2007). Later, deep neural networks are explored for the task. Chen et al. (Chen, Song, and Yang 2016) propose the first attempt to video denoising based on RNN. Vogels et al. (Vogels et al. 2018) design a kernel-predicting neural network for denoising Monte-Carlo-rendered sequences. Tassano et al. (Tassano, Delon, and Veit 2019) propose DVDnet by separating the denoising of a frame into two stages. More recently, Tassano et al. (Tassano, Delon, and Veit 2020) propose FastDVDnet to eliminate the dependence on motion estimation. Besides, some recent works focus on blind video denoising, e.g., (Ehret et al. 2019) and (Michele and Jan 2019).

Video illumination enhancement. Learning-based low-light image enhancement gains increasing attention recently (Yan et al. 2014, 2016; Lore, Akintayo, and Sarkar 2017; Cai, Gu, and Zhang 2018; Wang et al. 2019a; Moran et al. 2020; Guo et al. 2020). Wang et al. (Wang et al. 2019a) enhance photos by learning to estimate an illumination map. Sean et al. (Moran et al. 2020) learn spatial filters of various types for image enhancement. Also, unsupervised learning has been explored, e.g., Guo et al. (Guo et al. 2020) train a lightweight network to estimate pixel-wise and high-order curves for dynamic range adjustment. Yet, applying low-light image enhancement methods independently to individual frames will likely cause flickering, thus leading to research on methods for low-light videos, e.g., (Zhang et al. 2016; Lv et al. 2018; Jiang and Zheng 2019; Xue et al. 2019; Wang et al. 2019b; Chen et al. 2019). Zhang et al. (Zhang et al. 2016) adopt a perception-driven progressive fusion. Lv et al. (Lv et al. 2018) design a multi-branch network to extract multi-level features for stable enhancement.

Method

Architecture

To start, let us denote $\{X_t\}$ as the input LLN frames and $\{Y_t\}$ as the synthesized HNN frames, where t is the time index. Usually, we train the network with $\{X_t\}$ downsampled from the ground-truth frames $\{\widehat{Y}_t\}$ and we denote r as the downsampling rate. To obtain realistic and temporally-smooth videos, we consider N frames before and N frames after time t for estimating the target frame Y_t:

$$Y_t = f(X_{t'}, t' \in [t-N, t+N]). \quad (1)$$

Thus, the shape of the network input is $T \times H \times W \times C$, where $T = 2N+1$ and H, W, C are the height, width, channel size of the input video. Then, the shape of the output SR frame Y_t shall be $rH \times rW \times C$. Fig. 2 illustrates the network input, synthesized frame, and various components in our framework. Overall, our framework first synthesizes an intermediate HNN frame Z_t, then constructs residual image R_t to refine Z_t to generate the final output Y_t.

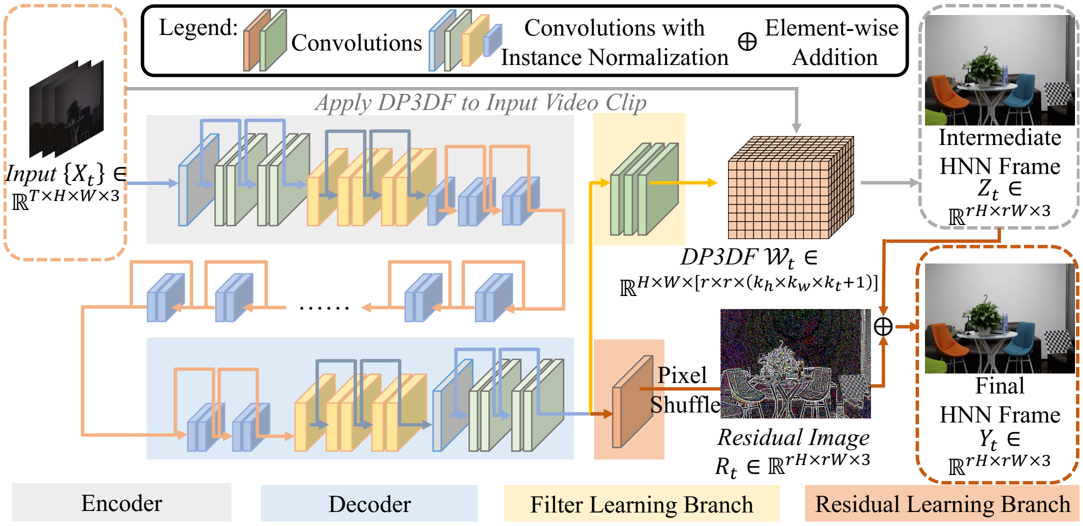

Legend:
Convolutions
Convolutions with Instance Normalization
⊕ Element-wise Addition
Input $\{X_t\} \in \mathbb{R}^{T\times H\times W\times 3}$
Apply DP3DF to Input Video Clip
DP3DF $\mathcal{W}_t \in \mathbb{R}^{H\times W\times[r\times r\times(k_h\times k_w\times k_t+1)]}$
Intermediate HNN Frame $Z_t \in \mathbb{R}^{rH\times rW\times 3}$
Pixel Shuffle
Residual Image $R_t \in \mathbb{R}^{rH\times rW\times 3}$
Final HNN Frame $Y_t \in \mathbb{R}^{rH\times rW\times 3}$
......
Encoder
Decoder
Filter Learning Branch
Residual Learning Branch

Figure 2: Overview of our framework. The encoder branch (grey area) extracts deep features from network input $\{X_t\}$ and the decoder branch (blue area) produces the output for learning the DP3DF and residual image R_t (the branch in the red area). Further, we learn the DF3DF (the branch in the yellow area) for synthesizing the intermediate HNN frame Z_t. Finally, we refine Z_t using R_t to produce the final output Y_t. The branch for learning the DP3DF and residual image share the same encoder-and-decoder backbone. see Fig. 3 for the details of how to apply DP3DF to a video.

We propose to first learn a new parametric representation called DP3DF for producing Z_t. To complete the mapping from $\{X_t\}$ to Z_t, DP3DF has the shape of $H \times W \times [r \times r \times (k_h \times k_w \times k_t + 1)]$, where k_h, k_w, and k_t are the dimensions (height, width, and time, respectively) of a 3D volume covered by DP3DF at each pixel in the network input and the "+1" is an additional component for illumination enhancement. Each pixel has $r \times r$ DP3DF kernels, each of size $k_h \times k_w \times k_t + 1$; see Fig. 3. For the enhancement of $X_t(p)$, where p denotes a pixel location, we sample a volume of $k_h \times k_w \times k_t$ pixels around p in X_t and then use the learned $r \times r$ kernels to produce $r \times r$ pixels for the original pixel at p. Besides, we normalize the $k_h \times k_w \times k_t$ elements in each kernel to be a sum of one for promoting smoothness in the results and suppressing the noise. The additional "one" dimension is for illumination adjustment.

DP3DF

Formulation. Our network predicts DP3DF $\mathcal{W}_t$ from $\{X_t\}$ and the filter learning branch output; see Fig. 2. DP3DF kernel $\mathcal{W}_t(p)$ associates with pixel $X_t(p)$ in X_t. Each $\mathcal{W}_t(p)$ can be decomposed into $r \times r$ kernels. Each kernel has shape $k_h \times k_w \times k_t + 1$ and can be decomposed into two parts: $W_t^b(p) \in \mathbb{R}^{k_h\times k_w\times k_t}$ (weights for SR and denoising) and $L_t^b(p) \in \mathbb{R}^1$ (weight for luminance adjustment), where $b \in [1, r \times r]$. Upsampled $r \times r$ pixels in Z_t are predicted as

$$\begin{aligned}\mathcal{Z}_W^{r_1\times r+r_2} &= \sum_{m=-s_h}^{s_h}\sum_{n=-s_w}^{s_w}\sum_{o=-s_t}^{s_t} W(m,n,o)\times X_{t+o}(i+m,j+n),\\ Z_t(i\times r+r_1, j\times r+r_2) &= \mathcal{Z}_W^{r_1\times r+r_2}\times L_t^{r_1\times r+r_2}(i,j),\end{aligned} \tag{2}$$

where $s_h = \frac{k_h-1}{2}$, $s_w = \frac{k_w-1}{2}$, $s_t = \frac{k_t-1}{2}$, $p = (i,j)$, $r_1 \in \{0, 1, ..., r-1\}$ and $r_2 \in \{1, 2, ..., r\}$, which together iterate over the $r \times r$ kernels in $W_t(p)$, and $W(m,n,o)$ denotes $W_t^{r_1\times r+r_2}(i,j)[m+s_h, n+s_w, o+s_t]$. Especially, the elements in $W_t^b(p)$ are normalized through Softmax, summing to one, whereas the elements in $L_t^b(p)$ are processed with the activation function of Sigmoid and we take reciprocals. The convolution with $W_t^b(p)$ gives an effect of spatial-temporal smoothing and helps achieve denoising. On the other hand, the multiplication with $L_t^b(p)$ adjusts the illumination and enhances the dark areas in the input frame. Also, the resulting $r \times r$ pixels produce a high-resolution frame from the low-resolution one.

Implementation. To learn the DP3DF, we adopt a network of an encoder-and-decoder structure. As shown in Fig. 2, the encoder has two downsampling layers, each with several residual blocks (He et al. 2016). These residual blocks can extract relevant features in each layer and use an instance normalization to reduce the gap between different types of videos. Then, we pass the features from the encoder through several residual blocks to produce the input feature of the decoder. Subsequently, the decoder adopts a pixel shuffle (Shi et al. 2016) for upsampling.

Residual Learning

To further enhance the performance, we adopt a residual learning branch (see the red area in Fig. 2) to learn a residual image R_t for enriching the final output with high-frequency details. Importantly, the residual image R_t is produced from multiple input frames $\{X_t\}$ rather than a single input frame, so sharing the same encoder-decoder structure with the main branch for predicting the DP3DF allows us to reduce the computational overhead. Finally, we combine the intermediate HNN frame Z_t with the learned residual image R_t to produce final output frame Y_t.

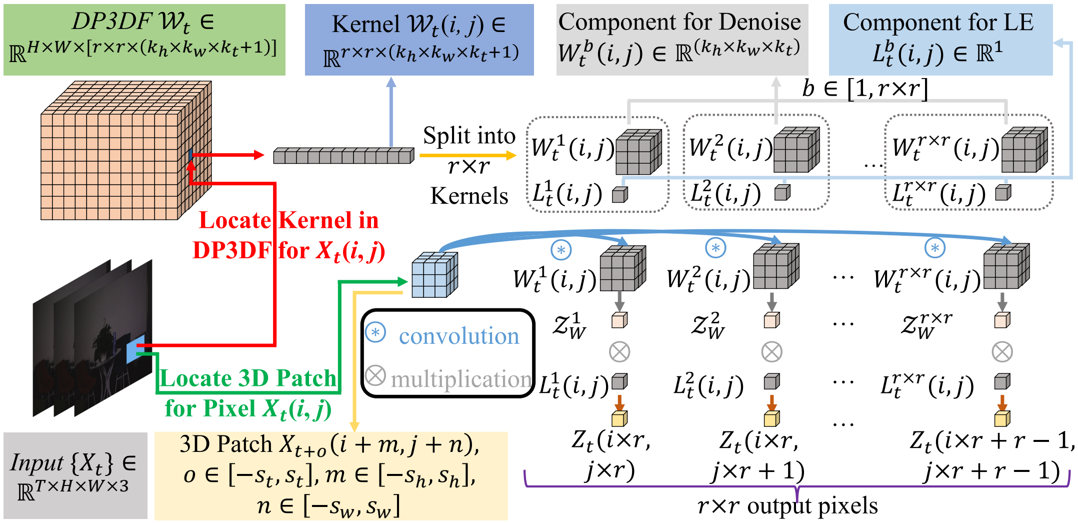

DP3DF $\mathcal{W}_t \in \mathbb{R}^{H\times W\times[r\times r\times(k_h\times k_w\times k_t+1)]}$
Kernel $\mathcal{W}_t(i,j) \in \mathbb{R}^{r\times r\times(k_h\times k_w\times k_t+1)}$
Component for Denoise $W_t^b(i,j) \in \mathbb{R}^{(k_h\times k_w\times k_t)}$
Component for LE $L_t^b(i,j) \in \mathbb{R}^1$
$b \in [1, r\times r]$
Split into $r\times r$ Kernels
$W_t^1(i,j)$
$L_t^1(i,j)$
$W_t^2(i,j)$
$L_t^2(i,j)$
$W_t^{r\times r}(i,j)$
$L_t^{r\times r}(i,j)$
Locate Kernel in DP3DF for $X_t(i,j)$
Locate 3D Patch for Pixel $X_t(i,j)$
convolution
multiplication
Z_W^1
Z_W^2
$Z_W^{r\times r}$
$Z_t(i\times r, j\times r)$
$Z_t(i\times r, j\times r+1)$
$Z_t(i\times r+r-1, j\times r+r-1)$
$r\times r$ output pixels
Input $\{X_t\} \in \mathbb{R}^{T\times H\times W\times 3}$
3D Patch $X_{t+o}(i+m, j+n)$, $o \in [-s_t, s_t]$, $m \in [-s_h, s_h]$, $n \in [-s_w, s_w]$

Figure 3: Illustrating how we apply the learned DP3DF to process an input video clip. For each pixel $X_t(i,j)$, we locate its 3D patch (green arrow) and its associated DP3DF kernel (red arrow), and then make use of the kernel components to process the 3D patch to produce $r \times r$ output pixels (yellow and blue arrows).

Loss Function

The overall loss has the following three parts.

(i) Reconstructing Z_t. First, we define an L_2 loss term for obtaining an accurate prediction of Z_t with the DP3DF:

$$\mathcal{L}_r = \|Z_t - \widehat{Y}_t\|, \tag{3}$$

where $\|\|$ is the L_2 norm, and all pixel channels in ground truth $\widehat{Y}_t$ and Z_t are normalized to [0, 1]. Such clip operation is effective for the training of illumination enhancement, eliminating invalid colors that are beyond the gamut and avoiding mistakenly darkening the underexposed regions.

(ii) Residual learning branch. Like Z_t, we define another reconstruction loss for the residual learning branch to generate the final output Y_t from Z_t and R_t:

$$\mathcal{L}_e = \|Y_t - \widehat{Y}_t\|. \tag{4}$$

(iii) Smoothness loss. Many works employ the smoothness prior for illumination enhancement, e.g., (Li and Brown 2014; Wang et al. 2019a), by assuming the illumination is locally smooth. Harnessing this prior in our framework has two advantages. It helps to not only reduce overfitting and improve the network's generalizability but also enhance the image contrast. For adjacent pixels, say p and q, with similar illumination values in a video frame, their contrast in the enhanced frame should be small; and vice versa. So, we define the smoothness loss on the predicted L_t^m as

$$\mathcal{L}_s = \textstyle\sum_t \sum_m \sum_p [v_t^p \times [\partial_x L_t^m(p)]^2 + u_t^p \times [\partial_y L_t^m(p)]^2], \tag{5}$$

where ∂_x and ∂_y are partial derivatives in horizontal and vertical directions, respectively, for the predicted L_t^m; v_t^p and u_t^q are spatially-varying (per-channel) smoothness weights expressed as

$$v_t^p = (\|\partial_x \mathcal{X}_t(p)\|^{1.2} + \epsilon)^{-1} \text{ and } u_t^p = (\|\partial_y \mathcal{X}_t(p)\|^{1.2} + \epsilon)^{-1}, \tag{6}$$

where $\mathcal{X}_t$ is the logarithmic image of X_t; and ϵ is a small constant (set to 0.0001) to prevent division by zero.

Overall loss. The overall loss $\mathcal{L}$ is

$$\mathcal{L} = \lambda_1 \mathcal{L}_r + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_e, \tag{7}$$

where λ_1, λ_2 and λ_3 are the loss weights.

Experiments

Datasets

We perform our evaluation on two public datasets with indoor and outdoor real-world videos: SMID (Chen et al. 2019) and SDSD (Wang et al. 2021). The videos in SMID are captured as static videos, in which the ground truths are obtained with a long exposure and the signal-to-noise ratio of the videos under the dark environment is extremely low. In this work, we explore the mapping from LLN to HNN frames in the sRGB domain. Thus, we follow the script provided by SMID (Chen et al. 2019) to convert the low-light videos from the RAW domain to the sRGB domain using rawpy's default ISP. On the other hand, SDSD is a dynamic video dataset collected through an electromechanical equipment, containing indoor and outdoor subsets. Also, we follow the official train-test split of SMID and SDSD.

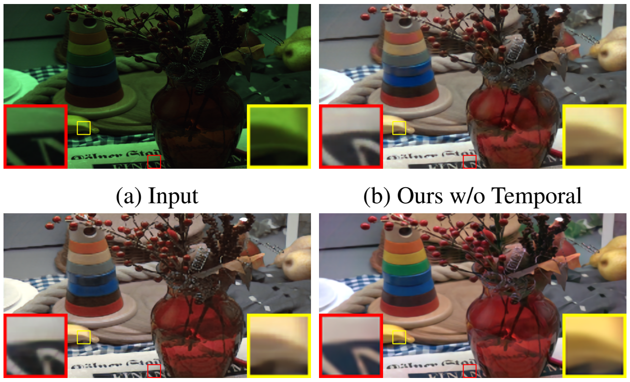
Implementation

We empirically set k_h=k_w=k_t=3 and number of frames T=3. Experiments on all datasets were conducted on the same network structure, whose backbone is an encoder-and-decoder structure; see Fig. 2. The encoder has three down-sampling layers with 64, 128, 256 channels, while the decoder has three up-sampling layers with 256, 128, 64 channels. The branches for learning the DP3DF and residual have three and one convolution layers, respectively.

We train all modules end-to-end with the learning rate initialized as 4e-4 for all layers (adapted by the cosine learning

	SMID		SDSD Indoor		SDSD Outdoor	
Methods	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours w/o T.	23.67	0.69	25.49	0.83	24.98	0.75
Ours w/o S.	22.84	0.63	24.87	0.78	24.02	0.71
Ours w/o R.	25.44	0.71	27.01	0.83	25.69	0.76
Ours	**25.73**	**0.73**	**27.11**	**0.85**	**25.80**	**0.77**

Table 1: The quantitative evaluation in the ablation study.


(a) Input

(b) Ours w/o Temporal

(c) Ours w/o Spatial

(d) Ours

Figure 4: Example visual samples in the ablation study.

scheduler); scale factor $r = 4$; batch size = 16; and patch size = 64×64. The patches are cropped randomly from the down-sampled low-resolution frame. We use Kaiming Initialization (He et al. 2015) to initialize the weights and Adam (Kingma and Ba 2014) for training with momentum set to 0.9. We implement our method using Python 3.7.7 and PyTorch 1.2.0 (Paszke et al. 2019), and ran all experiments on one NVidia TITAN XP GPU. PSNR and SSIM (Wang et al. 2004) are adopted for quantitative evaluation.

Ablation Study

We evaluate the major components in DP3DF on three ablated cases: (i) "w/o T." removes the property of the 3D filters by ignoring the temporal dimension and filtering only in the spatial dimensions; (ii) "w/o S." removes the spatial dimensions in DP3DF and applies filters only in the temporal dimension; and (iii) "w/o R." removes the branch of residual learning.

Table 1 summarizes the results, showing that all ablated cases are weaker than our full method. Especially, "w/o T." does not have the ability to incorporate information from the adjacent time frames and "w/o S." cannot obtain information from the adjacent pixels, thereby both having weaker performance. These two cases show the necessity of considering both the temporal and spatial dimensions in our 3D filter. Though "w/o R" leverages multiple frames as DP3DF, our full model still consistently achieves better results. Further, Fig. 4 shows some visual samples, revealing the apparent degradation caused by removing different components in our 3D filter.

	SMID		SDSD Indoor		SDSD Outdoor	
Methods	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BasicVSR	21.78	0.62	20.72	0.71	20.91	0.70
BasicVSR++	22.48	0.65	21.02	0.75	21.31	0.72
IconVSR	21.99	0.63	20.94	0.73	20.89	0.71
RBPN	24.87	0.72	23.47	0.80	22.46	0.74
Zooming	24.89	0.71	26.32	0.84	22.05	0.72
TGA	23.40	0.67	23.92	0.76	23.83	0.74
TDAN	24.65	0.70	24.00	0.80	22.57	0.74
PFNL	20.85	0.60	23.19	0.82	23.31	0.72
ToFlow	23.08	0.66	21.82	0.76	22.07	0.71
EDVR	24.50	0.70	25.00	0.83	23.37	0.75
Ours	**25.73**	**0.73**	**27.11**	**0.85**	**25.80**	**0.77**

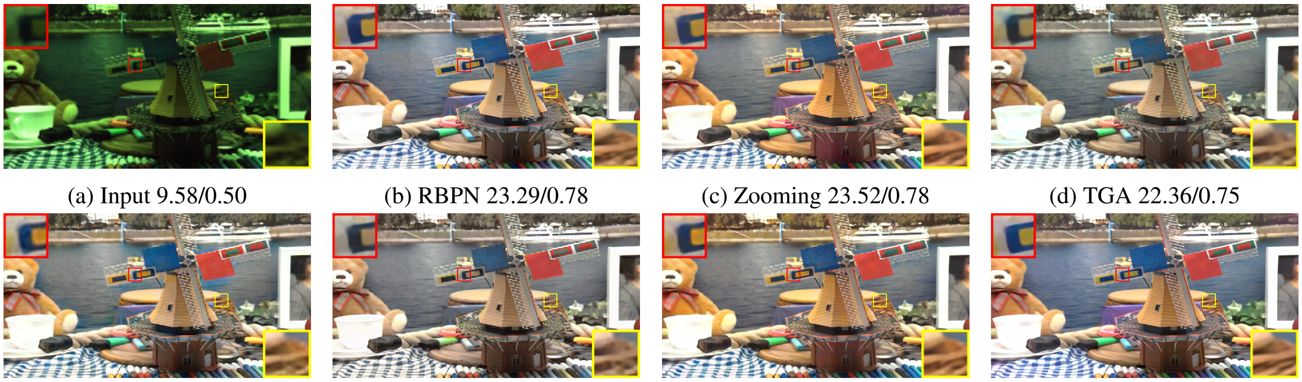
Table 2: Quantitative comparison with various SOTA SR methods on the SMID and SDSD datasets.

	SMID		SDSD Indoor		SDSD Outdoor	
Methods	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
F.+Z.	25.22	0.71	26.82	0.80	22.93	0.72
F.+TGA	23.97	0.70	24.15	0.78	24.11	0.74
F.+TDAN	24.95	0.71	24.30	0.76	23.55	0.70
T.+Z.	24.34	0.67	25.74	0.77	22.15	0.69
T.+TGA	23.07	0.66	23.65	0.72	23.48	0.70
T.+TDAN	24.12	0.68	23.69	0.71	23.01	0.67
F.+T.+Z.	23.97	0.74	26.54	0.81	24.41	0.73
F.+T.+TGA	24.57	**0.76**	25.31	0.78	25.01	0.75
F.+T.+TDAN	24.00	0.70	25.89	**0.87**	23.81	0.71
T.+F.+Z.	23.73	0.70	26.01	0.79	23.69	0.74
T.+F.+TGA	24.66	0.68	24.70	0.77	24.88	0.72
T.+F.+TDAN	24.21	0.71	24.88	0.81	23.35	0.70
Ours	**25.73**	0.73	**27.11**	0.85	**25.80**	**0.77**

Table 3: Comparison with baselines that combine SOTA video SR, denoise, and illumination enhancement networks. ··F." means FastDVDnet, "Z." denotes Zooming, "T" is TCE.

Comparison

Baselines. As far as we are aware of, there is no current work designed for directly mapping LNN videos to HNN videos. So, we choose the following two classes of works to compare with. First, we consider a rich collection of SOTA methods for video SR: BasicVSR (Chan et al. 2021), IconVSR (Chan et al. 2021), BasicVSR++ (Chan et al. 2022), RBPN (Haris, Shakhnarovich, and Ukita 2019), Zooming (Xiang et al. 2020), TGA (Isobe et al. 2020), TDAN (Tian et al. 2020), PFNL (Yi et al. 2019), ToFlow (Xue et al. 2019), and EDVR (Wang et al. 2019c). We trained them on each dataset with their released code. Second, we collectively use network models for video denoising, illumination enhancement, and SR in a cascaded manner: illumination enhancement+SR, denoising+SR, illumination enhancement+denoise+SR, and denoise+illumination enhancement+SR, where "+" indicates the order of using different networks. Here, we employ FastDVDnet (Tassano, Delon, and Veit 2020), a SOTA method for video denoising, and TCE (Zhang et al. 2021), a SOTA



(a) Input 9.58/0.50

(b) RBPN 23.29/0.78

(c) Zooming 23.52/0.78

(d) TGA 22.36/0.75

(e) TDAN 22.92/0.75

(f) ToFlow, 22.27/0.75

(g) EDVR 23.44/0.77

(h) Ours **24.54/0.79**

Figure 5: Qualitative comparison on SMID. Our result contains sharper details and more vivid colors. Score means PSNR/SSIM. Please zoom to view.

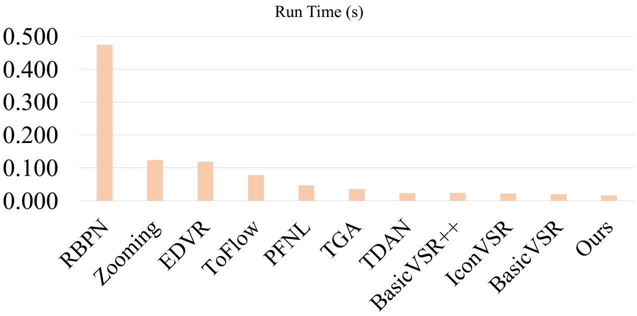


Figure 6: Quantitative comparisons between our framework and existing SOTA video SR methods in terms of method run time on input images of 960×512

method for video illumination enhancement, for use with various SOTA video SR methods.

Quantitative analysis. Table 2 shows the comparison results with the SOTA SR methods. From the table, we can see that our method *consistently* achieves the highest PSNR and SSIM for all the datasets. Especially, our PSNR values are higher than all others by a large margin. This superiority shows that our method has strong capability of enhancing LLN videos. Also, the right two columns show results on the SDSD indoor and outdoor subsets. These videos contain dynamic scenes, so they are very challenging to handle. Yet, our method is able to obtain high-quality results with top PSNR and SSIM for both subsets.

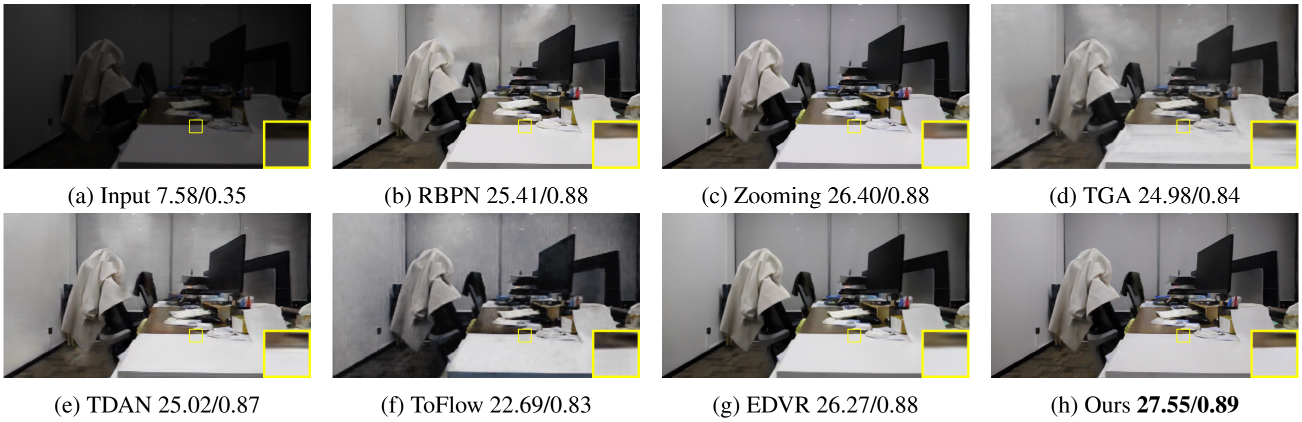
On the other hand, Table 3 summarizes the comparison results with baselines that collectively combine SOTA video denoising, illumination enhancement, and SR networks. Here, we trained each network (videos SR, illumination enhancement, and denoising) individually on the associated dataset. From Table 3, we can see that our method always produces top PSNR values for all three datasets and our SSIM values stay high compared with others.

Fig. 6 reports the run time of our method vs. the SOTA video SR methods. We ran all methods on Intel 2.6GHz CPU & TITAN XP GPU. From the figure, we can see that our method is efficient with very low running time.

Qualitative analysis. Next, we show visual comparisons with other methods. Fig. 5 shows the comparison on SMID. Overall, the results show two main advantages of our method over others. First, the result from our method has high contrast and clear details, as well as natural color constancy and brightness. Therefore, the frame processed by our method is more realistic than those by the others. Second, in regions with complex textures, it can be observed that our outputs have fewer artifacts. So, our result looks cleaner and sharper than those produced by the others. Further, these results demonstrate that our method can *simultaneously* achieve video SR, noise reduction, and illumination enhancement.

Figs. 7 and 8, respectively, show the visual comparisons on the SDSD indoor and outdoor subsets. Compared with the results of the baselines, our results are visually more appealing due to the explicit details, vivid colors, rational contrast, and plausible brightness. These results show the limitations of the existing approaches in converting LLN videos to HNN videos, and the superiority of our framework.

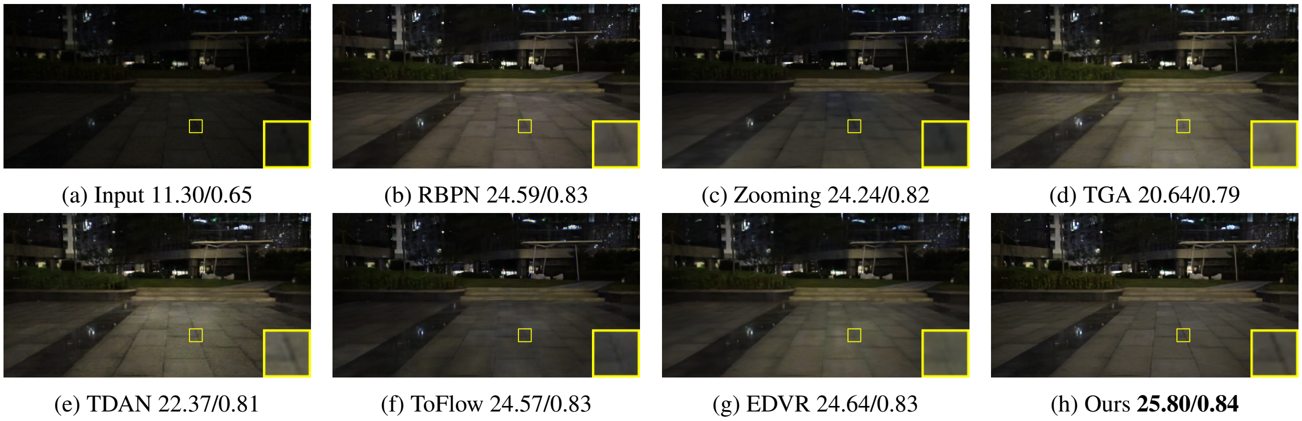
User study. Further, we conducted a large-scale user study with 80 participants (aged 18 to 52; 32 females and 48 males) to compare the perceptual quality of our method against various SOTA video SR approaches. In detail, we randomly selected 36 videos from the test sets of SMID and SDSD, and compared the results of different methods on these videos using an AB test. For each test video, our produced result is "Video A" whereas the result from some other baseline is "Video B." In the test, each participant had to simultaneously watch videos A and B (we avoid bias by randomizing the left-right presentation order when showing videos A and B in each AB-test task) and choose among three options: "I think Video A is better", "I think Video B is better", and "I cannot decide." Also, we asked the participants to make decisions based on the natural brightness, rich details, distinct contrast, and vivid color of the videos. For each participant, the number of tasks is 10 methods × 2 videos = 20, and it took around 30 minutes on average for each participant to complete the user study.



(a) Input 7.58/0.35 (b) RBPN 25.41/0.88 (c) Zooming 26.40/0.88 (d) TGA 24.98/0.84

(e) TDAN 25.02/0.87 (f) ToFlow 22.69/0.83 (g) EDVR 26.27/0.88 (h) Ours **27.55/0.89**

Figure 7: Qualitative comparison on indoor videos in the SDSD dataset Score means PSNR/SSIM. Our result contains sharper details and more vivid colors. Please zoom to view.



(a) Input 11.30/0.65 (b) RBPN 24.59/0.83 (c) Zooming 24.24/0.82 (d) TGA 20.64/0.79

(e) TDAN 22.37/0.81 (f) ToFlow 24.57/0.83 (g) EDVR 24.64/0.83 (h) Ours **25.80/0.84**

Figure 8: Qualitative comparison on outdoor videos in the SDSD dataset. Score means PSNR/SSIM. Our result contains sharper details and more vivid colors. Please zoom to view the sample frames.

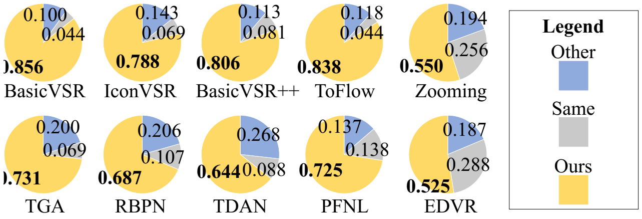


Figure 9: "Ours" is the percentage of test cases, in which the participant selected our results as better; "Other" is the percentage that another method was chosen to be better; and "Same" is the percentage that the user could not decide.

Fig. 9 summarizes the results of the user study, demonstrating that our results are more preferred by the participants over all the baselines. Also, we performed the statistical analysis by using the T-TEST function in MS Excel and found that the associated p-values in the comparison with the baseline methods are all smaller than 0.001, showing that the conclusion has a significant level of 0.001 statistically.

Conclusion

This paper presents a new approach for video super resolution. Our novel parametric representation, Deep Parametric 3D Filters (DP3DF), enables a direct mapping of LNN videos to HNN videos. It intrinsically incorporates local spatiotemporal information and achieves video SR simultaneously with denoising and illumination enhancement efficiently within a single encoder-and-decoder network. Besides, a dynamic residual frame can be jointly learned with the DP3DF, sharing the backbone and improving the visual quality of the results.

Extensive experiments were conducted on two real-world video datasets, SMID and SDSD, to show the effectiveness of our new approach. Both the quantitative and qualitative comparisons between our approach and current SOTA methods demonstrate our approach's consistent top performance. Further, an extensive user study with 80 participants was conducted to evaluate and compare the results in terms of human perception. Results also showed that our results consistently receive higher ratings than those from the baselines.

References

Arias, P.; and Morel, J.-M. 2018. Video denoising via empirical Bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision*.

Caballero, J.; Ledig, C.; Aitken, A.; Acosta, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Cai, J.; Gu, S.; and Zhang, L. 2018. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Trans. Image Process.*

Chan, K. C.; Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2021. BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Chan, K. C.; Zhou, S.; Xu, X.; and Loy, C. C. 2022. BasicVSR++: Improving Video Super-Resolution with Enhanced Propagation and Alignment. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Chen, C.; Chen, Q.; Do, M. N.; and Koltun, V. 2019. Seeing motion in the dark. In *Int. Conf. Comput. Vis.*

Chen, X.; Song, L.; and Yang, X. 2016. Deep RNNs for video denoising. In *Applications of Digital Image Processing XXXIX*.

Dabov, K.; Foi, A.; Katkovnik, V.; and Egiazarian, K. 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*

Ehret, T.; Davy, A.; Morel, J.-M.; Facciolo, G.; and Arias, P. 2019. Model-blind video denoising via frame-to-frame training. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Guo, C.; Li, C.; Guo, J.; Loy, C. C.; Hou, J.; Kwong, S.; and Cong, R. 2020. Zero-reference deep curve estimation for low-light image enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Haris, M.; Shakhnarovich, G.; and Ukita, N. 2019. Recurrent back-projection network for video super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Int. Conf. Comput. Vis.*

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Huang, Y.; Wang, W.; and Wang, L. 2017. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*

Isobe, T.; Li, S.; Jia, X.; Yuan, S.; Slabaugh, G.; Xu, C.; Li, Y.-L.; Wang, S.; and Tian, Q. 2020. Video super-resolution with temporal group attention. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Jiang, H.; and Zheng, Y. 2019. Learning to see moving objects in the dark. In *Int. Conf. Comput. Vis.*

Jo, Y.; Oh, S. W.; Kang, J.; and Kim, S. J. 2018. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Li, Y.; and Brown, M. S. 2014. Single image layer separation using relative smoothness. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Lim, B.; and Lee, K. M. 2017. Deep recurrent ResNet for video super-resolution. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*.

Lore, K. G.; Akintayo, A.; and Sarkar, S. 2017. LLNet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*.

Lv, F.; Lu, F.; Wu, J.; and Lim, C. 2018. MBLLEN: Low-Light Image/Video Enhancement Using CNNs. In *Brit. Mach. Vis. Conf.*

Maggioni, M.; Boracchi, G.; Foi, A.; and Egiazarian, K. 2012. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Trans. Image Process.*

Michele, C.; and Jan, V. G. 2019. ViDeNN: Deep blind video denoising. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*

Moran, S.; Marza, P.; McDonagh, S.; Parisot, S.; and Slabaugh, G. 2020. DeepLPF: Deep Local Parametric Filters for Image Enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*

Sajjadi, M. S.; Vemulapalli, R.; and Brown, M. 2018. Frame-recurrent video super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A. P.; Bishop, R.; Rueckert, D.; and Wang, Z. 2016. Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Adv. Neural Inform. Process. Syst.*

Tao, X.; Gao, H.; Liao, R.; Wang, J.; and Jia, J. 2017. Detail-revealing deep video super-resolution. In *Int. Conf. Comput. Vis.*

Tassano, M.; Delon, J.; and Veit, T. 2019. DVDnet: A fast network for deep video denoising. In *IEEE International Conference on Image Processing (ICIP)*.

Tassano, M.; Delon, J.; and Veit, T. 2020. FastDVDnet: Towards real-time deep video denoising without flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Tian, Y.; Zhang, Y.; Fu, Y.; and Xu, C. 2020. TDAN: Temporally-deformable alignment network for video super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Vogels, T.; Rousselle, F.; McWilliams, B.; Röthlin, G.; Harvill, A.; Adler, D.; Meyer, M.; and Novák, J. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics*.

Wang, L.; Guo, Y.; Lin, Z.; Deng, X.; and An, W. 2018. Learning for video super-resolution through HR optical flow estimation. In *ACCV*.

Wang, R.; Xu, X.; Fu, C.-W.; and Jia, J. 2021. Seeing Dynamic Scene in the Dark: High-Quality Video Dataset with Mechatronic Alignment. In *Int. Conf. Comput. Vis.*

Wang, R.; Zhang, Q.; Fu, C.-W.; Shen, X.; Zheng, W.-S.; and Jia, J. 2019a. Underexposed Photo Enhancement Using Deep Illumination Estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Wang, W.; Chen, X.; Yang, C.; Li, X.; Hu, X.; and Yue, T. 2019b. Enhancing Low Light Videos by Exploring High Sensitivity Camera Noise. In *Int. Conf. Comput. Vis.*

Wang, X.; Chan, K. C.; Yu, K.; Dong, C.; and Chen, C. L. 2019c. EDVR: Video restoration with enhanced deformable convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*

Xiang, X.; Tian, Y.; Zhang, Y.; Fu, Y.; Allebach, J. P.; and Xu, C. 2020. Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time Video Super-Resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Xue, T.; Chen, B.; Wu, J.; Wei, D.; and Freeman, W. T. 2019. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.*

Yan, J.; Lin, S.; Sing, B. K.; and Tang, X. 2014. A learning-to-rank approach for image color enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Yan, Z.; Zhang, H.; Wang, B.; Paris, S.; and Yu, Y. 2016. Automatic photo adjustment using deep neural networks. *ACM Trans. Graph.*

Yi, P.; Wang, Z.; Jiang, K.; Jiang, J.; and Ma, J. 2019. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *Int. Conf. Comput. Vis.*

Zhang, F.; Li, Y.; You, S.; and Fu, Y. 2021. Learning Temporal Consistency for Low Light Video Enhancement From Single Images. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Zhang, Q.; Nie, Y.; Zhang, L.; and Xiao, C. 2016. Underexposed Video Enhancement via Perception-Driven Progressive Fusion. *IEEE Trans. Vis. Comput. Graph.*