On the Generalization of BasicVSR++ to Video Deblurring and Denoising

–Technical Report–

Kelvin C.K. Chan **Shangchen Zhou** **Xiangyu Xu** **Chen Change Loy**
S-Lab, Nanyang Technological University
{chan0899, s200094, xiangyu.xu, ccloy}@ntu.edu.sg

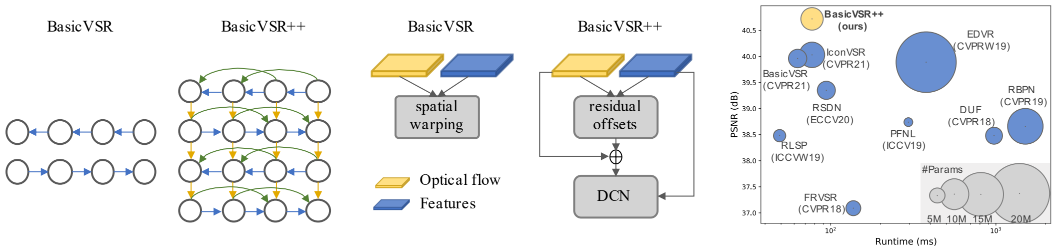
Abstract

The exploitation of long-term information has been a long-standing problem in video restoration. The recent BasicVSR and BasicVSR++ have shown remarkable performance in video super-resolution through long-term propagation and effective alignment. Their success has led to a question of whether they can be transferred to different video restoration tasks. In this work, we extend BasicVSR++ to a generic framework for video restoration tasks. In tasks where inputs and outputs possess identical spatial size, the input resolution is reduced by strided convolutions to maintain efficiency. With only minimal changes from BasicVSR++, the proposed framework achieves compelling performance with great efficiency in various video restoration tasks including video deblurring and denoising. Notably, BasicVSR++ achieves comparable performance to Transformer-based approaches with up to 79% of parameter reduction and 44$\times$ speedup. The promising results demonstrate the importance of propagation and alignment in video restoration tasks beyond just video super-resolution. Code and models are available at `https://github.com/ckkelvinchan/BasicVSR_PlusPlus`.

1 Introduction

Long-term propagation and effective alignment has been shown essential in video super-resolution [10, 31, 34]. In our previous work [3], we summarize existing video super-resolution pipelines into four components, namely *alignment*, *propagation*, *aggregation*, and *upsampling*. Based on the decomposition, **BasicVSR** is proposed with simple designs. Without dedicated components for video super-resolution, BasicVSR demonstrates superior performance and outperforms state of the arts with improved efficiency. The simplicity and effectiveness of BasicVSR demonstrate the potential of the recurrent framework for video super-resolution.

Motivated by the success, we further improve BasicVSR by replacing the primitive propagation and alignment modules with more sophisticated designs. We propose **BasicVSR++** with two modifications to employ long-term information more effectively, as shown in Fig. 1. First, second-order grid propagation is proposed to more aggressively transmit information across video frames: (1) The second-order connection extends the conventional approach of nearest-frame propagation and distributes information also to the second-next timestamp. In such a way, gradient vanishing can be partially alleviated, and information can be propagated to further timestamps. (2) Grid propagation refines the intermediate features through propagation. Specifically, instead of propagating the features along each direction once, the features are circulated back-and-forth for feature refinement, exploiting long-term information. Second, BasicVSR++ goes beyond flow-based alignment in BasicVSR and adopts *flow-guided deformable alignment*, combining the motion prior from optical flow and the flexibility of deformable alignment [4]. The main idea is to adopt optical flow as the base offsets, and residual offsets are learned. The combined offsets are then used in deformable convolution [8, 38]



(a) Propagation (b) Alignment (c) Performance Gain

Figure 1: **Architecture and performance of BasicVSR++. (a)** Second-order grid propagation in BasicVSR++ allows a more effective propagation of features. **(b)** Flow-guided deformable alignment in BasicVSR++ provides a means for more robust feature alignment across misaligned frames. **(c)** BasicVSR++ outperforms existing state-of-the-art super-resolution methods including its predecessor, BasicVSR, while maintaining efficiency.

for feature alignment. Such a design alleviates the training instability of deformable alignment and improve the alignment accuracy.

As it is a common goal to exploit temporal information in video restoration, we hypothesize that the success of BasicVSR++ is not limited to video super-resolution. In this work, we extend our scope to a more video restoration tasks and introduce a generic framework built on BasicVSR++. For tasks where inputs and outputs possess the same resolution, we reduce the input resolution by strided convolutions to maintain efficiency. In addition to video super-resolution [24] and compressed video enhancement [35], we show that BasicVSR++ is generalizable to video deblurring and denoising, achieving promising performance with high efficiency.

2 Video Restoration Framework

The original BasicVSR++ assumes that the input resolution is $4\times$ smaller than the output resolution. In this work, we extend BasicVSR++ to a generic video restoration framework. In cases where the input resolution is equal to output resolution (*e.g.*, deblurring, denoising), we introduce the following two modifications:

1. To improve efficiency, we apply strided convolution to the input frames to reduce the spatial resolution.
2. To further reduce the computational cost, we downsample the input frames to the reduced resolution for optical flow computation.

With these designs, most of the computations are performed in the low-resolution feature space, substantially improving efficiency. The remaining operations follow that of BasicVSR++. We refer readers to the original paper [5] for more details.

3 Experiments

In this section, we discuss the performance in *video deblurring* and *video denoising*. For video super-resolution and compressed video enhancement, we refer readers to our original paper [5] and the NTIRE 2021 challenge report [35], respectively.

3.1 Video Deblurring

Settings. We mostly follow the settings of the original BasicVSR++, except that we increase the number of residual blocks in each of the four propagation branches from 7 to 15. In addition, strided convolutions are used to reduce the spatial resolution by 4 times. We adopt Adam optimizer [14] and Cosine Annealing scheme [19]. The initial learning rate of the main network and the flow network

Table 1: **Quantitative comparison on DVD [25] (Video Deblurring).** Green and blue colors indicate the best and the second-best performance, respectively.

	EDVR [31]	Tao *et al.* [27]	Su *et al.* [25]	DBLRNet [36]	STFAN [37]	Xiang *et al.* [33]	TSP [23]	Suin *et al.* [26]	ARVo [16]	**BasicVSR++**
PSNR	28.51	29.98	30.01	30.08	31.15	31.68	32.13	32.53	32.80	**34.28**
SSIM	0.864	0.884	0.888	0.885	0.905	0.916	0.827	0.947	0.935	**0.951**

Table 2: **Quantitative comparison on GoPro [21] (Video Deblurring).** Green and blue colors indicate the best and the second-best performance, respectively.

	RDN [32]	Kim *et al.* [12]	EDVR [31]	Su *et al.* [25]	STFAN [37]	Nah *et al.* [22]	Tao *et al.* [27]	TSP [23]	Suin *et al.* [26]	**BasicVSR++**
PSNR	25.19	26.82	26.83	27.31	28.59	29.97	30.29	31.67	32.10	**34.01**
SSIM	0.779	0.825	0.843	0.826	0.861	0.895	0.901	0.928	**0.960**	0.952

Table 3: **Complexity comparison on DVD [25] (Video Deblurring).** Green and blue colors indicate the best and the second-best performance, respectively. Runtime and FLOPs are measured on an RTX 2080 Ti GPU with spatial resolutions 1280×720 and 240×240, respectively.

	EDVR [31]	Su *et al.* [25]	STFAN [37]	TSP [23]	**BasicVSR++**
PSNR (dB)	28.51	30.01	31.15	32.13	**34.28**
Params (M)	23.60	15.30	**5.37**	16.19	9.76
FLOPs (G)	159.2	38.7	**35.4**	357.9	37.6
Runtime (ms)	268.5	133.2	145.9	579.7	**130.5**

Input | STFAN | TSP | **BasicVSR++ (ours)** | GT

Figure 2: **Qualitative comparison on DVD [25] (Video Deblurring).** Only BasicVSR++ is able to restore the the word "DORIC".

are set to 1×10^{-4} and 2.5×10^{-5}, respectively. The weights of the flow network are fixed during the first 5,000 iterations. The batch size is 8 and the patch size of input frames is 256×256. We use Charbonnier loss [6] since it better handles outliers and improves the performance over the conventional ℓ_2-loss [15]. The number of iterations is set to 600,000 and 200,000 when training on DVD [25] and GoPro [21], respectively. During training, 30 frames are used as inputs. For testing, we take the full video sequence as inputs to explore information from all video frames for restoration. The detailed configurations can be found in `https://github.com/ckkelvinchan/BasicVSR_PlusPlus` and MMEditing [7].

Quantitative Comparison. As shown in Table 1 and Table 2, BasicVSR++ outperforms existing works by a large margin in terms of PSNR on both DVD [25] and GoPro [21]. Notably, it outperforms the second-best methods by **1.48 dB** and **1.91 dB** on DVD and GoPro, respectively. Furthermore, thanks to the downsampling mechanism at the input end, BasicVSR++ possesses high efficiency (see

Table 4: **Quantitative comparison (PSNR/SSIM) on DAVIS [13] (Video Denoising).** The improvements over existing works increase with the noise level σ. **Green** and blue colors indicate the best and the second-best performance, respectively. Δ denotes the performance gain over PaCNet [30]. Note that PaCNet trains different networks for different noise levels.

	VBM4D [20]	VNLB [1]	DVDnet [28]	FastDVDnet [29]	VNLNet [9]	PaCNet [30]	**BasicVSR++**	Δ
σ=10	37.58/-	38.85/-	38.13/0.9657	38.71/0.9672	39.56/0.9707	39.97/0.9713	**40.13/0.9754**	0.16/0.0003
σ=20	33.88/-	35.68/-	35.70/0.9422	35.77/0.9405	36.53/0.9464	37.10/0.9470	**37.41/0.9598**	0.31/0.0128
σ=30	31.65/-	33.73/-	34.08/0.9188	34.04/0.9167	-	35.07/0.9211	**35.74/0.9457**	0.67/0.0246
σ=40	30.05/-	32.32/-	32.86/0.8962	32.82/0.8949	33.32/0.8996	33.57/0.8969	**34.49/0.9321**	0.92/0.0352
σ=50	28.80/-	31.13/-	31.85/0.8745	31.86/0.8747	-	32.39/0.8743	**33.45/0.9179**	0.83/0.0343
Average	32.39/-	34.34/-	34.52/0.9195	31.64/0.9188	-	35.62/0.9221	**36.24/0.9462**	0.62/0.0241

Table 5: **Quantitative comparison (PSNR/SSIM) on Set8 [29] (Video Denoising).** The improvements over existing works increase with the noise level σ. **Green** and blue colors indicate the best and the second-best performance, respectively. Δ denotes the performance gain over PaCNet [30]. Note that PaCNet trains different networks for different noise levels.

	VBM4D [20]	VNLB [1]	DVDnet [28]	FastDVDnet [29]	VNLNet [9]	PaCNet [30]	**BasicVSR++**	Δ
σ=10	36.05/-	37.26/-	36.08/0.9510	36.44/0.9540	**37.28/0.9606**	37.06/0.9590	36.83/0.9574	-0.45/-0.0003
σ=20	32.18/-	33.72/-	33.49/0.9182	33.43/0.9196	34.08/0.9273	33.94/0.9247	**34.15/0.9319**	0.07/0.0005
σ=30	30.00/-	31.74/-	31.68/0.8862	31.68/0.8889	-	32.05/0.8921	**32.57/0.9095**	0.52/0.0174
σ=40	28.48/-	30.39/-	30.46/0.8564	30.46/0.8608	30.72/0.8622	30.70/0.8623	**31.42/0.8889**	0.70/0.0266
σ=50	27.33/-	29.24/-	29.53/0.8289	29.53/0.8351	-	29.66/0.8349	**30.49/0.8692**	0.83/0.0343
Average	30.81/-	32.47/-	32.29/0.8881	32.31/0.8917	-	32.68/0.8946	**33.09/0.9114**	0.41/0.0168

Table 6: **Runtime comparison on Set8 [29] (Video Denoising).** Notably, BasicVSR++ is 546$\times$ faster than PaCNet with 0.41 dB improvement in PSNR. **Green** and blue colors indicate the best and the second-best performance, respectively. Runtime is measured on an RTX 2080 Ti GPU.

	VBM4D [20]	VNLB [1]	DVDnet [28]	FastDVDnet [29]	VNLNet [9]	PaCNet [30]	**BasicVSR++**
PSNR (dB)	30.81	32.47	32.29	32.31	-	32.68	**33.09**
Runtime (s)	420.0	156.0	2.51	**0.08**	1.65	35.24	**0.08**

Table 8 for the FLOPs comparison). In particular, as shown in Table 3, BasicVSR++ has the fastest speed while achieving the highest PSNR.

Qualitative Comparison. In Fig. 2 we compare our BasicVSR++ with STFAN [37] and TSP [23]. Through aggregating long-term information from the video sequence, BasicVSR++ is able to restore the blurry texts, whereas other methods failed to recover the details.

3.2 Video Denoising

Settings. The settings follow that for deblurring in DVD, except that (1) the number of frames used in training is 25 and (2) the batch size is reduced to 7. The detailed configurations can be found in `https://github.com/ckkelvinchan/BasicVSR_PlusPlus` and MMEditing [7].

Quantitative Comparison. From Table 4 and Table 5, it is observed that BasicVSR++ outperforms existing works with a much higher efficiency. For example, as shown in Table 6, BasicVSR++ is **546**$\times$ faster than PaCNet [30] while having 0.62 dB and 0.41 dB improvements on DAVIS [13] and Set8 [29], respectively. Interestingly, we find that the improvements over previous state of the arts increase with the noise level σ. We conjecture that long-term information is more important for low-quality videos, where most information is lost due to the severe noise.

Qualitative Comparison. In Fig. 3 we show an example of σ=50. With severe noise, VBM4D [20] and FastDVDnet [29] are unable to restore the numbers faithfully. In contrast, the effective propagation and alignment of BasicVSR++ lead to a better output with more details revealed.

3.3 Comparison to Transformer-Based Approaches

Recently, Transformer-based methods [2, 17, 18] have shown competitive performance in various video restoration tasks, including super-resolution, deblurring, and denoising. In this section, we

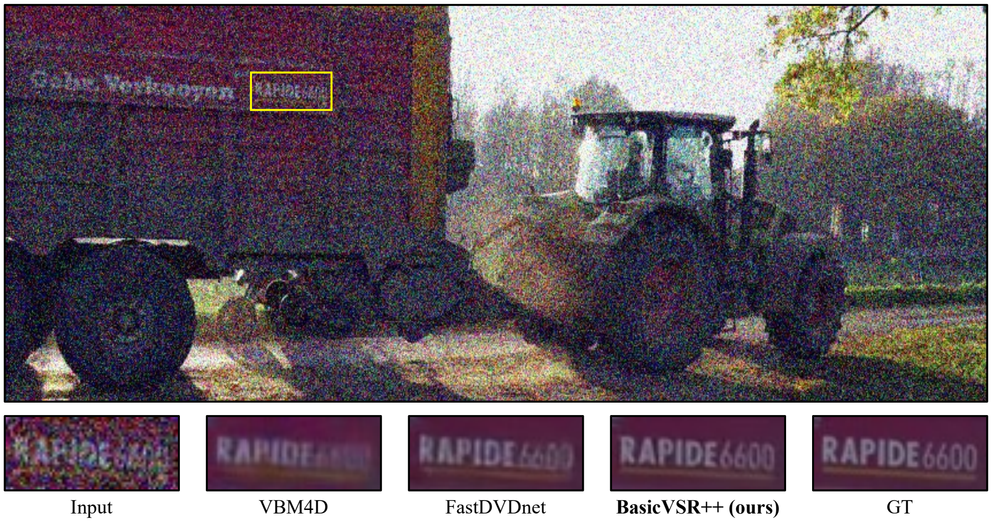


Figure 3: **Qualitative comparison on Set8 [13] (Video Denoising).** Only BasicVSR++ is able to restore the number "6600".

Table 7: **Comparison (PSNR (dB) / Params (M) / Runtime (ms)) with Transformer-based methods.** BasicVSR++ achieves comparable performance to Transformer-based methods with better efficiency. **Green** and blue colors indicate the best and the second-best performance, respectively. Runtime is measured on an RTX 2080 Ti GPU with an output resolution of 1280×720.

	VSRT [2]	FGST [18]	VRT [17]	**BasicVSR++**
Super-Resolution (REDS4)	31.06 / 32.6 / 4312	-	32.19 / 35.6 / 243	**32.39** / **7.3** / **98**
Deblurring (DVD)	-	33.36 / **9.7** / 247	34.27 / 18.3 / 220	**34.28** / 9.8 / **131**
Denoising (DAVIS)	-	-	**34.36** / 18.3 / 220	33.45 / **9.8** / **131**

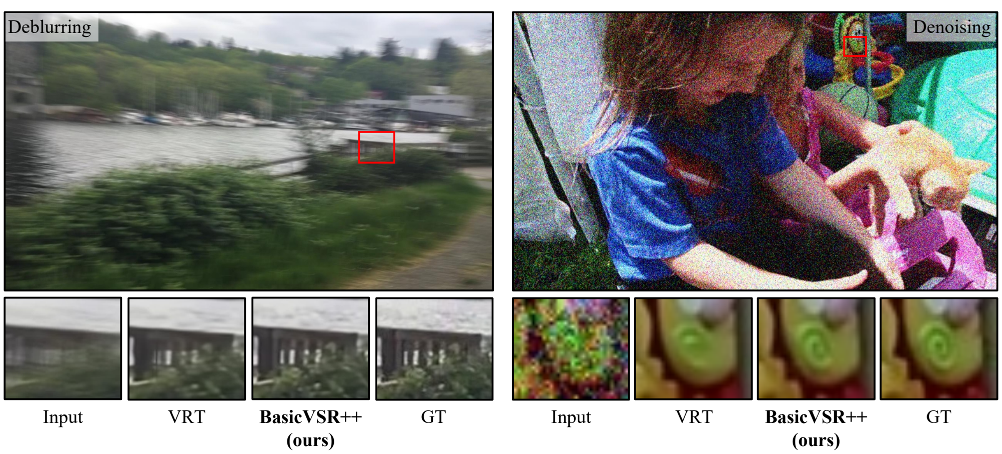


Figure 4: **Qualitative comparison with VRT.** Despite its smaller complexity, BasicVSR++ is able to achieve comparable results to VRT.

compare BasicVSR++ (recurrent framework) to the Transformer-based methods. The comparison is shown in Table 7.

Transformer-based methods achieve remarkable performance in the aforementioned tasks. For example, VRT [17] achieves comparable performance to BasicVSR++ in deblurring and outperforms BasicVSR++ in denoising. However, these methods generally require large network complexity to

Table 8: **Comparison between different implementations of BasicVSR++ on Set8.** With $4\times$ downsampling, our implementation maintains high efficiency. FLOPs is measured with an output resolution of 960×540.

Implementation	σ=10	σ=20	σ=30	σ=40	σ=50	FLOPs (G)
Huang *et al.* [11]	37.27	34.25	32.55	31.28	30.32	3402
Ours	36.83	34.15	32.57	31.42	30.49	303
Δ	-0.44	-0.10	0.02	0.14	0.17	91%

achieve good performance. For instance, for super-resolution, VSRT [2] and VRT [17] consist of over 30M parameters, which are about 5 times of BasicVSR++. In contrast, BasicVSR++ exploits long-term information through a recurrent network, achieving a promising performance-parameter-speed tradeoff. As shown in Fig. 4, through exploiting long-term information, BasicVSR++ is able to reconstruct sharper edges and clearer details, achieving results highly similar to the ground truth.

3.4 Performance-Speed Tradeoff with Input Downsampling

Our video restoration framework naturally provides a tradeoff between performance and speed. On the one hand, downsampling the inputs leads to an increased efficiency by potentially results in weaker learning power. On the other hand, operating in the input resolution leads to a lower speed due to the computations in high-resolution. One could determine the reduction factor of the spatial resolution based on the task nature.

In Table 8 we compare our implementation (with $4\times$ downsampling) and that from Huang *et al.* [11] (without downsampling) to understand the tradeoff. By shifting the computations to lower resolutions, our implementation maintains high efficiency. However, the performance in cases of mild noises is inferior to that from Huang *et al.*. In addition, our performance improves with the noise level. This shows that the downsampling operation may harm the performance in cases of mild degradations, where rich information can be acquired from the input frames. More thorough exploration of this tradeoff is left as our future work.

References

[1] Pablo Arias and Jean-Michel Morel. Towards a bayesian video denoising method. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2015.

[2] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021.

[3] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. BasicVSR: The search for essential components in video super-resolution and beyond. In *CVPR*, 2021.

[4] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. In *AAAI*, 2021.

[5] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 2022.

[6] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994.

[7] MMEditing Contributors. MMEditing: OpenMMLab Image and Video Editing Toolbox, 2022.

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[9] Axel Davy, Thibaud Ehret, and Gabriele Facciolo. Non-local video denoising by CNN. *arXiv preprint arXiv:1811.12758*, 2018.

[10] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019.

[11] Cong Huang, Jiahao Li, Bin Li, Dong Liu, and Yan Lu. Neural compression-based feature learning for video restoration. In *CVPR*, 2022.

[12] Tae Hyun Kim and Kyoung Mu Lee. Generalized video deblurring for dynamic scenes. In *CVPR*, 2015.

[13] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, 2018.

[14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[15] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017.

[16] Dongxu Li, Chenchen Xu, Kaihao Zhang, Xin Yu, Yiran Zhong, Wenqi Ren, Hanna Suominen, and Hongdong Li. ARVo: Learning all-range volumetric correspondence for video deblurring. In *CVPR*, 2021.

[17] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. VRT: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

[18] Jing Lin, Yuanhao Cai, Xiaowan Hu, Haoqian Wang, Youliang Yan, Xueyi Zou, Henghui Ding, Yulun Zhang, Radu Timofte, and Luc Van Gool. Flow-guided sparse transformer for video deblurring. *arXiv preprint arXiv:2201.01893*, 2022.

[19] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[20] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising using separable 4D nonlocal spatiotemporal transforms. In *Image Processing: Algorithms and Systems*, 2011.

[21] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.

[22] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *CVPR*, 2019.

[23] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *CVPR*, 2020.

[24] Sanghyun Son, Suyoung Lee, Seungjun Nah, Radu Timofte, Kyoung Mu Lee, Kelvin C.K. Chan, et al. NTIRE 2021 challenge on video super-resolution. In *CVPRW*, 2021.

[25] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, 2017.

[26] Maitreya Suin and A. N. Rajagopalan. Gated spatio-temporal attention-guided video deblurring. In *CVPR*, 2021.

[27] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.

[28] Matias Tassano, Julie Delon, and Thomas Veit. DVDnet: A fast network for deep video denoising. In *ICIP*, 2019.

[29] Matias Tassano, Julie Delon, and Thomas Veit. FastDVDnet: Towards real-time deep video denoising without flow. In *CVPR*, 2020.

[30] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Patch craft: Video denoising by deep modeling and patch matching. In *ICCV*, 2021.

[31] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019.

[32] Patrick Wieschollek, Michael Hirsch, Bernhard Scholkopf, and Hendrik Lensch. Learning blind motion deblurring. In *ICCV*, 2017.

[33] Xinguang Xiang, Hao Wei, and Jinshan Pan. Deep video deblurring using sharpness features from exemplars. *TIP*, 2020.

[34] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019.

[35] Ren Yang, Radu Timofte, Jing Liu, Yi Xu, Xinjian Zhang, Minyi Zhao, Shuigeng Zhou, Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, et al. NTIRE 2021 challenge on quality enhancement of compressed video: Methods and results. In *CVPRW*, 2021.

[36] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *TIP*, 2018.

[37] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *ICCV*, 2019.

[38] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.