BasicVSR++: Improving Video Super-Resolution with Enhanced Propagation and Alignment –Supplementary Material–

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

1. Network Architecture

We use pretrained SPyNet [8] as our flow network. The number of residual blocks for the initial feature extraction is set to 5, and the number of residual blocks for each propagation branch is set to 7. The feature channel is set to 64.

The architecture of our second-order deformable alignment is highly similar to the first-order counterpart (Fig. 3 in the main paper). The only difference is that the pre-aligned features and optical flows from different timesteps are concatenated, and passed to the offset estimation module $\mathcal{C}^o$ and mask estimation module $\mathcal{C}^m$. Their architectures are detailed in Table 1. We set the DCN kernel size to 3 and the number of deformable groups to 16. Codes will be released.

Table 1. **Architectures of $\mathcal{C}^o$ and $\mathcal{C}^m$.** The two modules share the first six layers. They can be implemented as a stack of convolutions followed by a channel-splitting. The arguments in the convolution layer are *input channels*, *output channels*, and *kernel size*, respectively.

Layer	$\mathcal{C}^o$	$\mathcal{C}^m$
1.	conv(196, 64, 3)	
2.	LeakyReLU(0.1)	
3.	conv(64, 64, 3)	
4.	LeakyReLU(0.1)	
5.	conv(64, 64, 3)	
6.	LeakyReLU(0.1)	
7.	conv(64, 288, 3)	conv(64, 144, 3)

2. Experimental Settings

Datasets. Two widely-used datasets are adopted for training: REDS [7] and Vimeo-90K [10]. For REDS, following BasicVSR [1], we use REDS4[1] as our test set and REDSval4[2] as our validation set. The remaining clips are used for training. We use Vid4 [5], UDM10 [11], and Vimeo-90K-T [10] as test sets along with Vimeo-90K.

Degradations. All models are tested with $4\times$ downsampling using two degradations – Bicubic (BI) and Blur Downsampling (BD). For BI, the MATLAB function `imresize` is used for downsampling. For BD, we blur the ground-truth by a Gaussian filter with σ=1.6, followed by a subsampling every four pixels.

Training Settings. We adopt Adam optimizer [3] and Cosine Annealing scheme [6]. When trained on REDS, the initial learning rate of the main network and the flow network are set to 1×10^{-4} and 2.5×10^{-5}, respectively. The total number of iterations is 600K, and 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 LR frames is 64×64. We use Charbonnier loss [2] since it better handles outliers and improves the performance over the conventional ℓ_2-loss [4]. During training, 30 LR frames are used as inputs. Since Vimeo-90K contains only seven frames per sequence, networks trained solely on Vimeo-90K may not be able to capture long-term dependencies. Therefore, we initialize the model using the weights trained on REDS when trained on Vimeo-90K. The number of finetune iterations is 300K.

Test Settings. We take the full video sequence as inputs to explore information from all video frames for restoration.

3. Limitations of Recurrent Framework

In this section, we will discuss the limitations of BasicVSR++ and more generally the recurrent framework, to provide insights for future works.

Long Training Time. Since BasicVSR++ and other recurrent networks are intended to exploit long-term information, they are usually trained with a long sequence, such as 15 or 30 frames. As a result, when compared to sliding-window methods such as EDVR [9], the training time of recurrent VSR networks is longer.

Large Memory Footprint. In bidirectional recurrent networks, intermediate features of the entire sequence has to

[1]Clips 000, 011, 015, 020 of REDS training set.
[2]Clips 000, 001, 006, 017 of REDS validation set.

be cached. Therefore, the memory footprint would increase with the length of sequence. Nevertheless, this can be ameliorated with some hardware workarounds such as caching the features in CPU.

4. Qualitative Comparisons

In this section, we provide additional qualitative comparisons on REDS4 [7], UDM10 [11], Vimeo-90K [10], and Vid4 [5]. From the examples, we see that BasicVSR++ is able to restore the fine details, leading to plausible results. A video demo is also provided in the submitted zip file.

References

[1] 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. 1

[2] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 1

[3] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[4] 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. 1

[5] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *TPAMI*, 2014. 1, 2, 4

[6] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 1

[7] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019. 1, 2

[8] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 1

[9] 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. 1, 3

[10] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019. 1, 2, 4

[11] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, 2019. 1, 2, 3

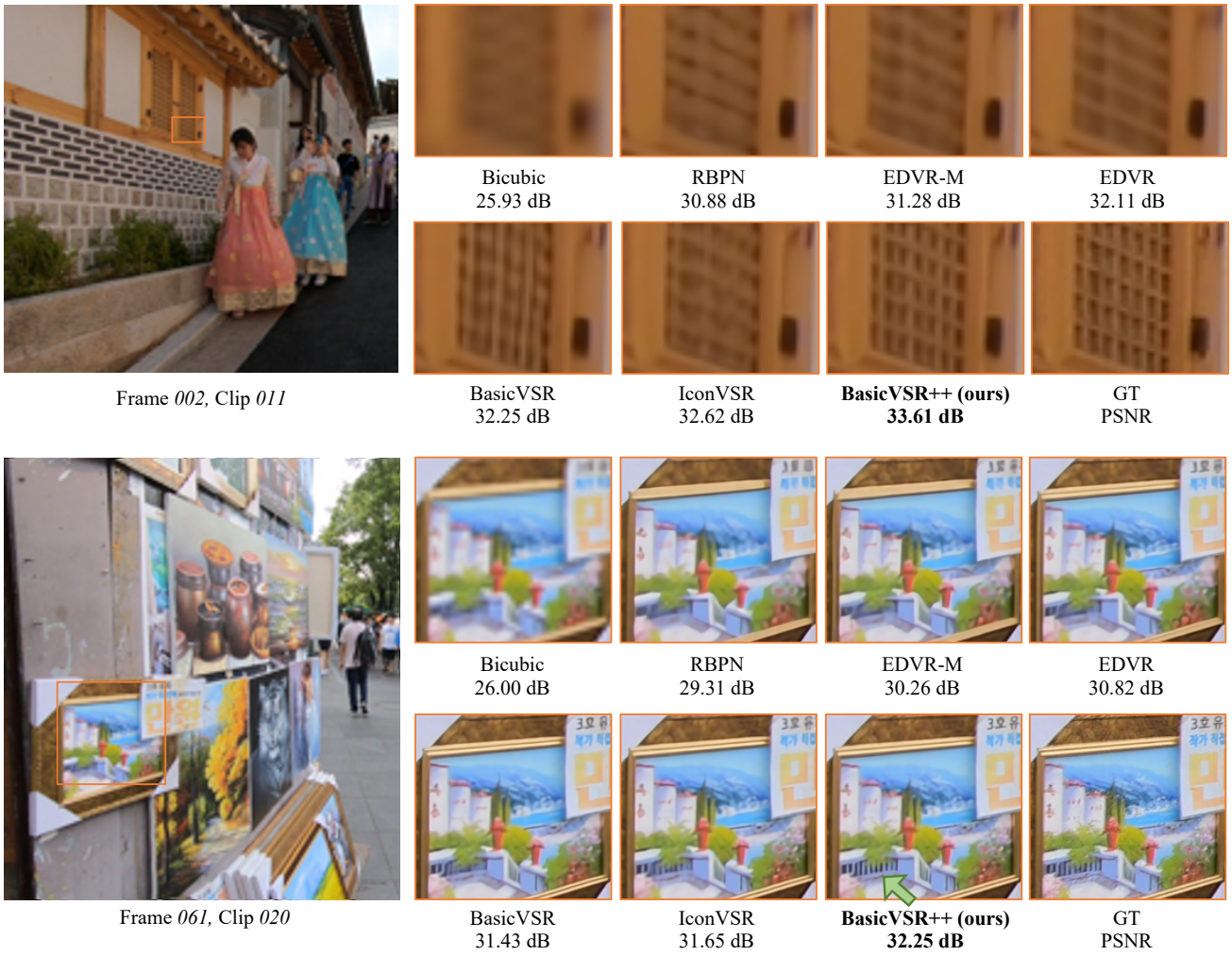

Figure 1. Qualitative comparison on REDS4 [9].

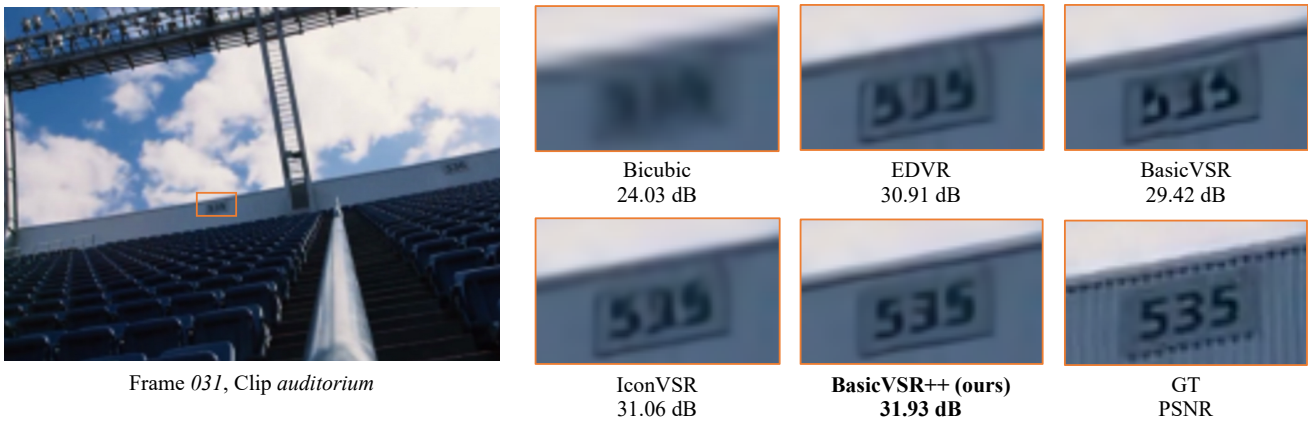

Figure 2. Qualitative comparison on UDM10 [11].

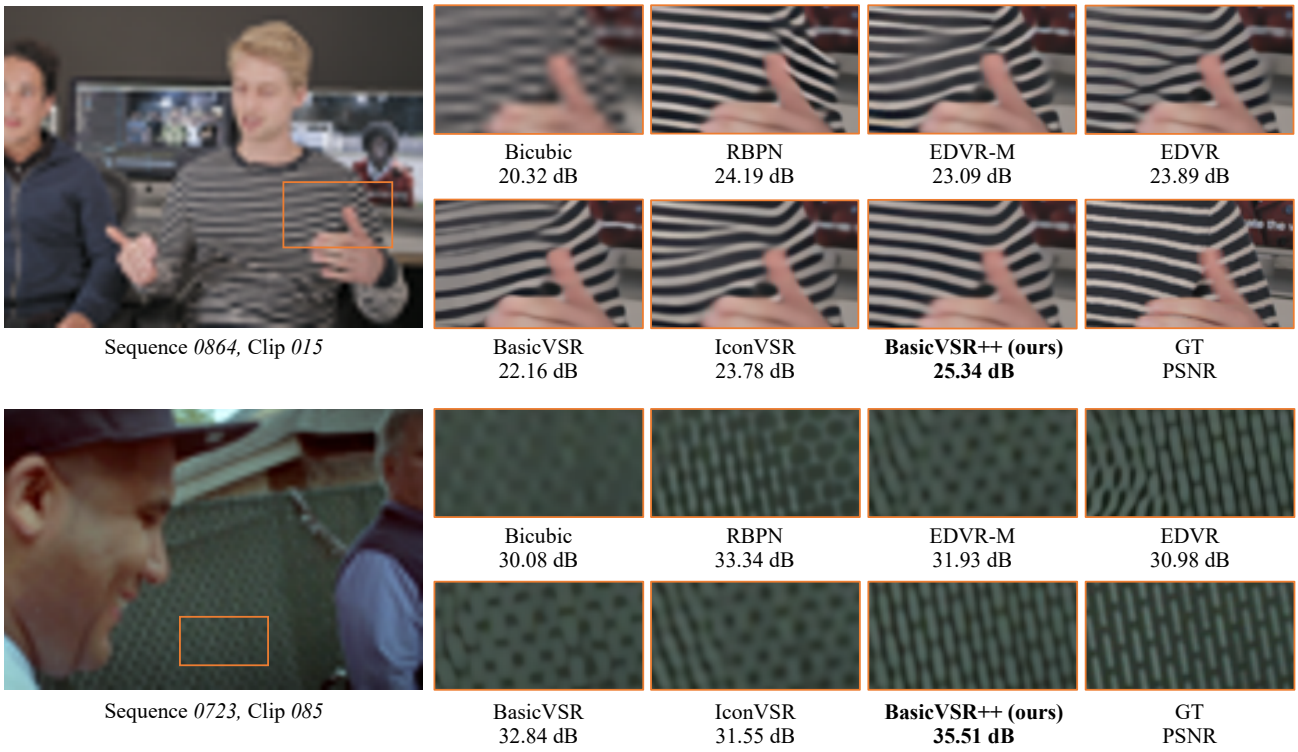

Figure 3. Qualitative comparison on Vimeo-90K-T [10].

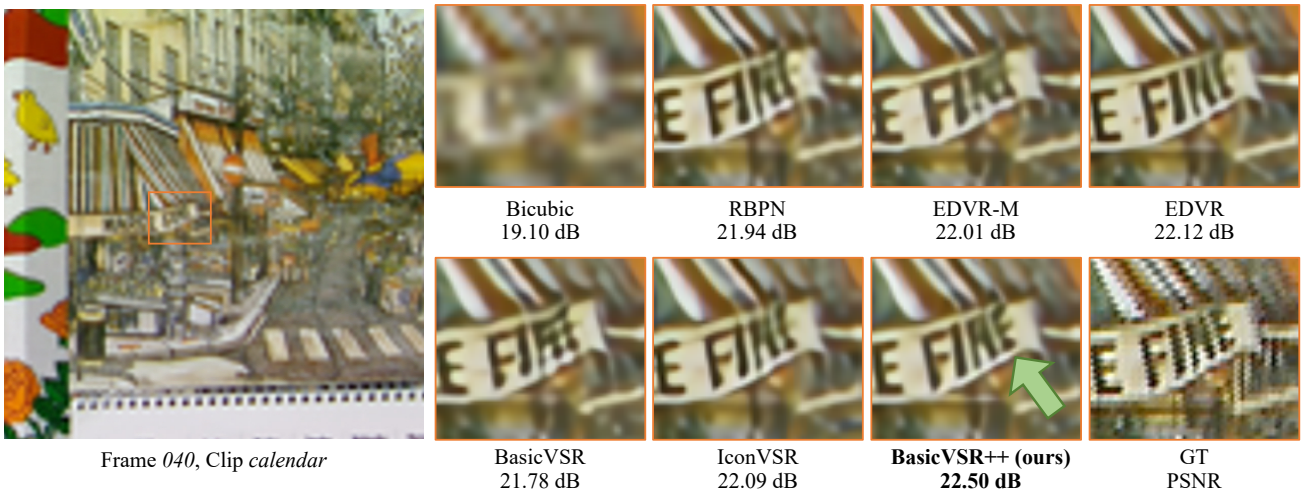

Figure 4. Qualitative comparison on Vid4 [5].