BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond Supplementary Material

Kelvin C.K. Chan[1] Xintao Wang[2] Ke Yu[3] Chao Dong[4,5] Chen Change Loy[1]
[1]S-Lab, Nanyang Technological University [2]Applied Research Center, Tencent PCG
[3]CUHK – SenseTime Joint Lab, The Chinese University of Hong Kong
[4]Shenzhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[5]SIAT Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society
{chan0899, ccloy}@ntu.edu.sg xintao.wang@outlook.com
yk017@ie.cuhk.edu.hk chao.dong@siat.ac.cn

1. Architecture and Experimental Settings

Architecture. In all our models, we adopt SPyNet [8] as our flow estimator because of its simplicity and efficiency. We use 30 residual blocks in each propagation branch. The feature channel is set to 64. In IconVSR, we adopt EDVR-M[1] [10] as the additional feature extractor since it maintains a good balance between efficiency and quality. The complexity of the components are summarized in Table 1. BasicVSR and IconVSR share the same flow estimator and main network. The main network is a lightweight network, consisting of only 4.9M parameters. The flow estimator and feature extractor are fine-tuned together with the main network. In all our experiments, every five frames are selected as keyframes. Note that the feature extractor is applied to keyframes only. Therefore, the computational burden brought by it is insignificant.

Table 1. **Model complexity of BasicVSR and IconVSR.**

	BasicVSR	IconVSR
Flow Estimator	1.4M	1.4M
Main Network	4.9M	4.9M
Feature Extractor	-	2.4M
Total	6.3M	8.7M

Datasets. We consider two widely-used datasets for training: REDS [7] and Vimeo-90K [11]. For REDS, following [10], we use the REDS4 dataset[2] as our test set. We additionally define REDSval4[3] as our validation set. The remaining clips are used for training. We use Vid4 [5], UDM10 [12], and Vimeo-90K-T [11] as test sets along with Vimeo-90K.

Experimental Settings. When training on REDS, we use a sequence of 15 frames as inputs, and loss is computed for the 15 output images. When training on Vimeo-90K, we temporally augment the sequence by flipping the original input sequence to allow longer propagation. In other words, we train with a sequence of 14 frames. During inference, we take the whole video sequence as input.

We adopt Adam optimizer [3] and Cosine Annealing scheme [6]. The initial learning rates of the feature extractor and flow estimator are set to 1×10^{-4} and 2.5×10^{-5}, respectively. The learning rate for all other modules is set to 2×10^{-4}. The total number of iterations is 300K, and the weights of the feature extractor and flow estimator are fixed during the first 5,000 iterations. The batch size is 8 and the patch size of input LR frames is 64×64.

Loss Function. We use Charbonnier loss [1] since it better handles outliers and improves the performance over the conventional ℓ_2 loss [4]:

$$\mathcal{L} = \frac{1}{N}\sum_{i=0}^{N}\rho(y_i - z_i), \tag{1}$$

where $\rho(x) = \sqrt{x^2+\epsilon^2}$, ϵ=1×10^{-8}, z_i denotes the ground-truth HR frame, and N denotes to the number of pixels.

Degradations. We train and test our models with $4\times$ downsampling using two degradations – Bicubic (BI) and Blur Downsampling (BD) [2, 9]. For BI, we use the MATLAB function `imresize` for downsampling. For BD, we blur

[1]A lightweight version of EDVR.
[2]Clips 000, 011, 015, 020 of REDS training set.
[3]Clips 000, 001, 006, 017 of REDS validation set.

the ground-truths by a Gaussian filter with σ=1.6, followed by a subsampling every four pixels.

Implementation. We implement our models with PyTorch and train the models using two NVIDIA Tesla V100 GPUs. Codes will be made publicly available.

2. Qualitative Results

2.1. Comparison with State of the Arts

In this section, we provide additional qualitative comparisons on REDS4 [7], Vimeo-90K [11], Vid4 [5], and UDM10 [12]. In Fig. 1 to Fig. 4, it is observed that BasicVSR and IconVSR successfully produce outputs with finer details and sharper edges. Furthermore, with the proposed information-refill and coupled propagation, IconVSR further improves the quality of the outputs.

2.2. BasicVSR vs IconVSR

In Fig. 5, we provide additional visual comparison of BasicVSR and IconVSR to demonstrate the effectiveness of our proposed components. We see that (1) information-refill improves the output quality on the fine regions, where alignment is error-prone, and (2) coupled propagation leads to sharper edges by better employing the long-term information in the sequence.

References

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

[2] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *ECCV*, 2020. 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*, pages 5835–5843, 2017. 1

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

[6] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 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, 3

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

[9] Mehdi S M Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, 2018. 1

[10] 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

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

[12] 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, 5

Figure 1. Qualitative comparison on REDS [7].

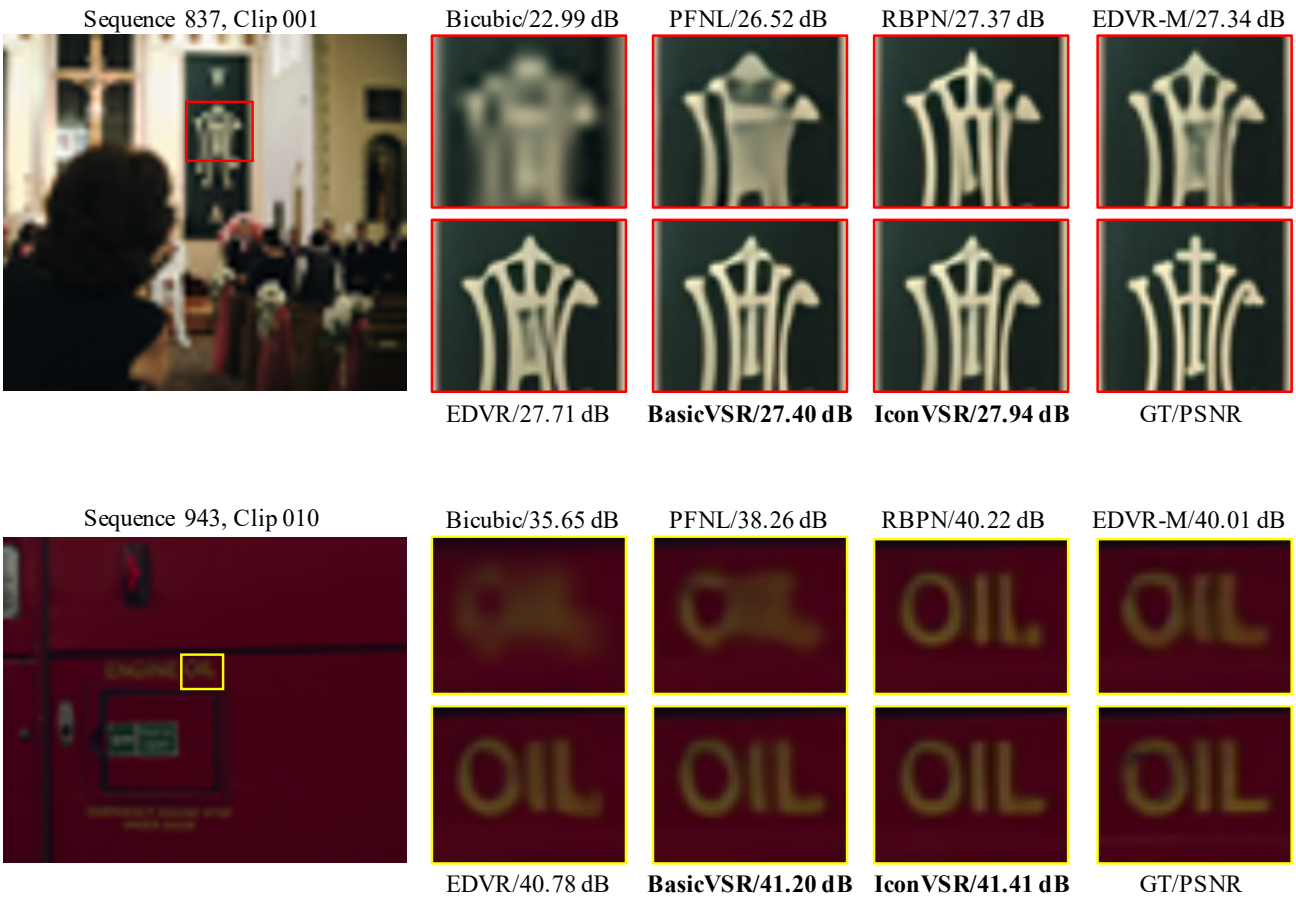


Figure 2. Qualitative comparison on Vimeo-90K [11].

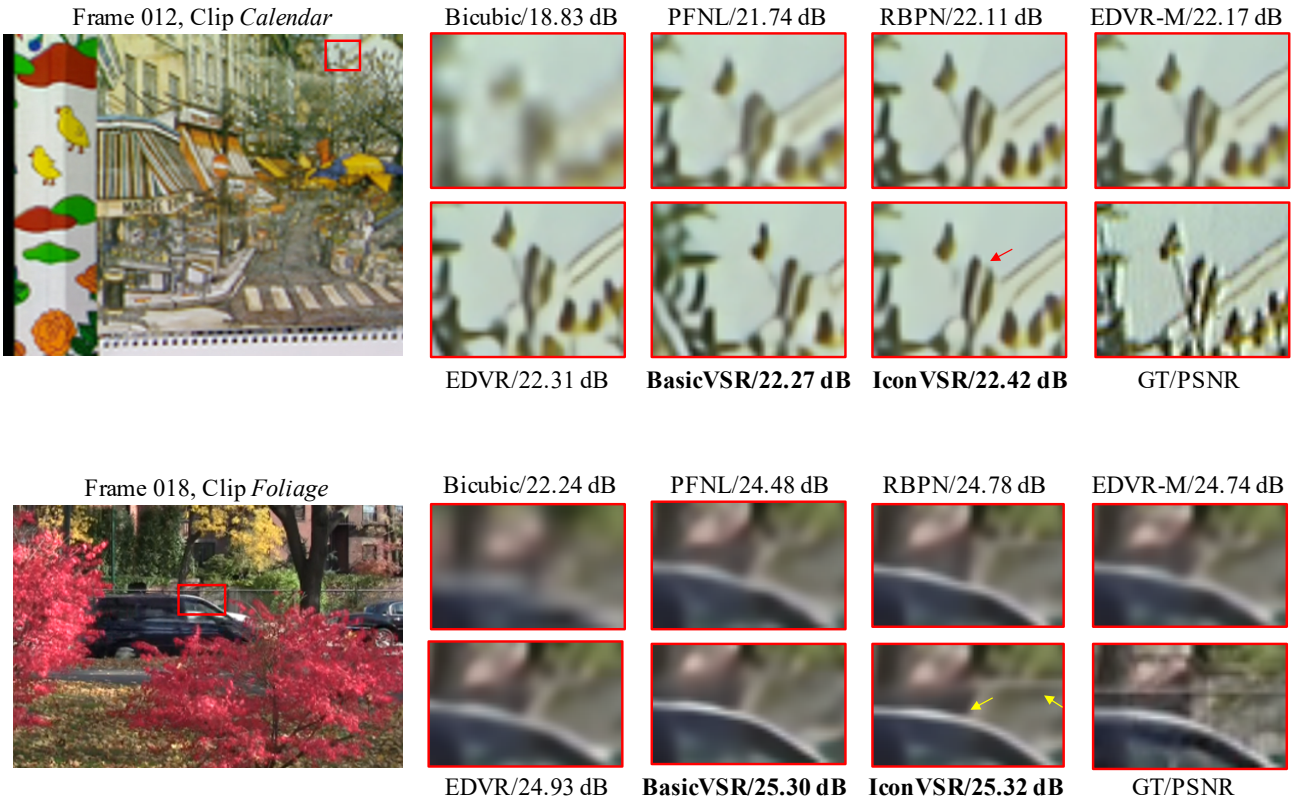


Figure 3. Qualitative comparison on Vid4 [5].

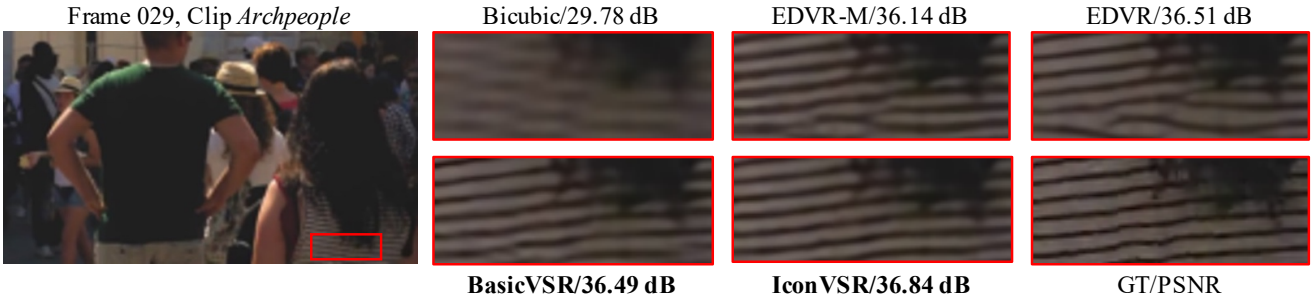
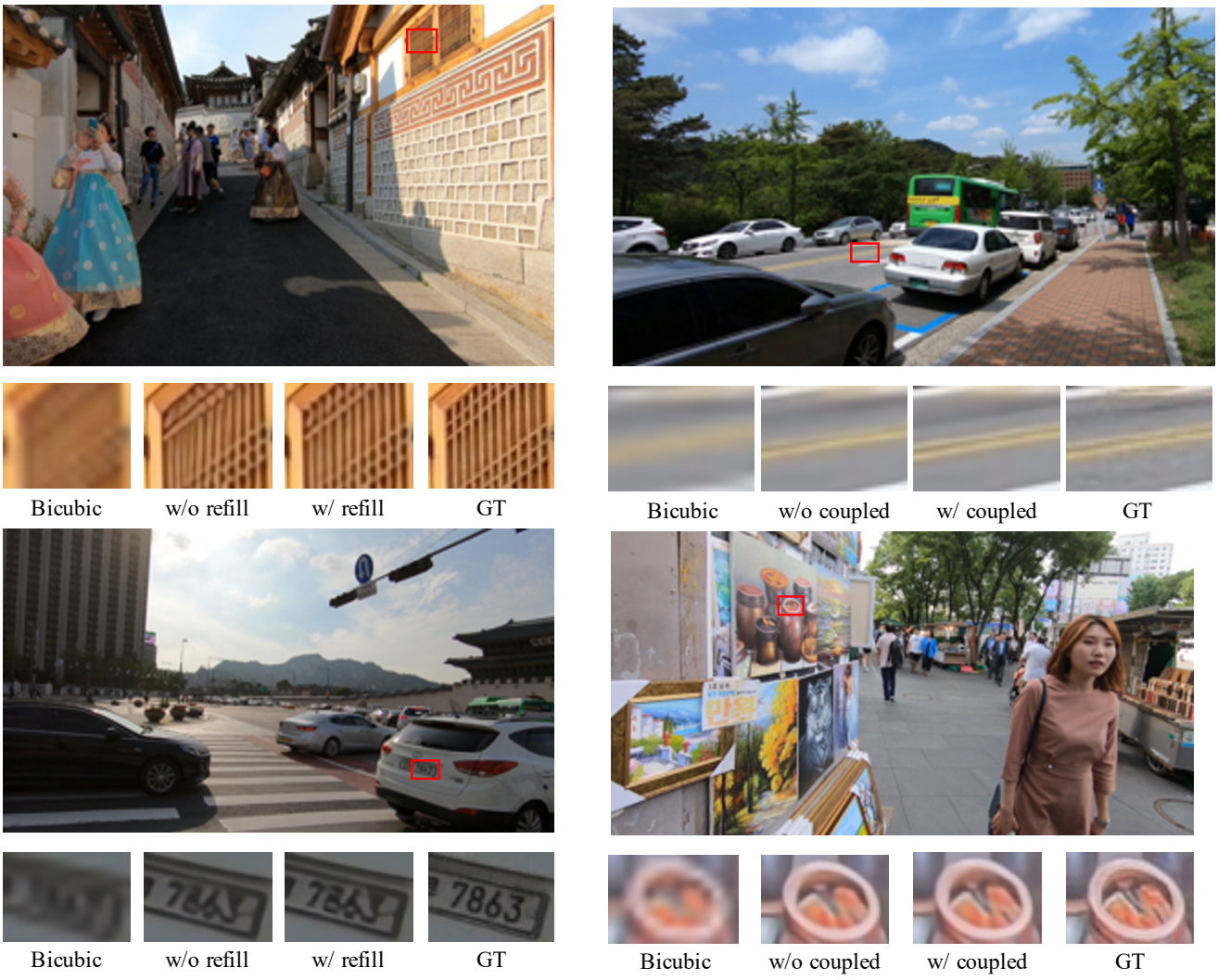


Figure 4. Qualitative comparison on UDM10 [12].



(a) Information-Refill

(b) Coupled Propagation

Figure 5. **Ablation of IconVSR.** With *information-refill* and *coupled propagation*, IconVSR produces outputs with details and sharper edges.