Supplementary Material – Fast Online Video Super-Resolution with Deformable Attention Pyramid

Dario Fuoli[1] Martin Danelljan[1] Radu Timofte[1,2] Luc Van Gool[1,3]
[1]Computer Vision Lab, ETH Zürich, Switzerland [2]CAIDAS, University of Würzburg, Germany [3]KU Leuven, Belgium
{dario.fuoli, martin.danelljan, vangool}@vision.ee.ethz.ch, radu.timofte@uni-wuerzburg.de

We provide additional visual analysis of the sampling locations predicted by our model, analysis of offset location statistics, analysis of temporal information propagation in our recurrent cell, a visual inspection and discussion of reverse evaluation results, additional results on Vimeo-90K and further details of our architecture.

1. Visual Analysis of Sampling Locations

We analyze the predicted sampling locations of our deformable attention pyramid (DAP) visually in Fig. 1. A randomly selected subset of pixel-dense offsets are shown for key/value sampling, overlaid on top of frame x_{t-1}. These locations are marked with crosses. Their corresponding pixel-dense query location is shown in frame x_t on the right hand side in Fig. 1. The points are marked in the same color. A visual inspection reveals offset predictions that match its corresponding location with high precision. The network learns to attend to offsets that are spread around the point of interest for increased robustness.

2. Offset Analysis

In order to investigate the pixel-dense offsets predicted by our DAP module, we plot the statistics from each video sequence on REDS with 2 types of histograms in Fig. 2. The top row provides 1D histograms to show the distribution of offset magnitudes in each sequence. The offset magnitudes are different, depending on the video content. The predicted offsets in sequence 0 are significantly smaller compared to others, which we attribute to more distant objects in the scene after visual inspection. The closer objects appear in a video, the larger their offsets grow. The second row in Fig. 2 contains 2D histograms, showing the prominent offset directions and magnitudes. The plots hint at the type of movement in each sequence, e.g. sequence 11 has larger offsets than sequence 0, indicating larger camera movement or more close-up content. Similar arguments can be made about sequence 20. Sequence 15 shows a unique horizontal offset pattern, which is a consequence of a camera pan with horizontally moving objects. There is a concentrated direction of offsets in one direction, and another set of offsets in the opposite direction. These differences are likely caused by opposing movement of background and foreground objects in the scene when the camera pans.

3. Analysis of Temporal Information Propagation

We investigate the evolution of PSNR in each sequence of REDS (test set) in Fig. 3. In order to show the importance of temporal aggregation from previous frames, we plot PSNR curves with different starting points, i.e. we initialize an empty hidden state at regular intervals (every 10th frame) and from there evaluate our model until the end of the whole sequence.

We investigate the model DAP-128, which was trained on 15 frames, i.e. the model we use for comparison to state of the art in the paper. Our proposed recurrent temporal information aggregation mechanism (DAP) efficiently leverages temporal information to improve super-resolution of a single frame. The effect is significant, as shown by the steep initialization curves exposed by subsequent intervals. In some cases - depending on the content - it takes more than 15 frames to reach the previously started model's performance with initial gaps of several dB in PSNR. Thus, the experiments show the benefits of having access to past information over a long temporal range, efficiently realized by our DAP aggregation mechanism through the hidden state in our recurrent cell.

4. Reverse Evaluation - Visual Results

We show the qualitative differences between forward and reverse evaluation in Fig. 4. The quantitative differences are investigated in the paper, we list these results again for reference in Tab. 1. The performance gain is attributed to the camera's motion direction as explained in the paper. If an object is first visible in higher resolution (larger), the network can leverage this higher-resolution information about the object in lower resolutions (smaller) The performance gain is clearly visible in 3 out of 4 sequences from the

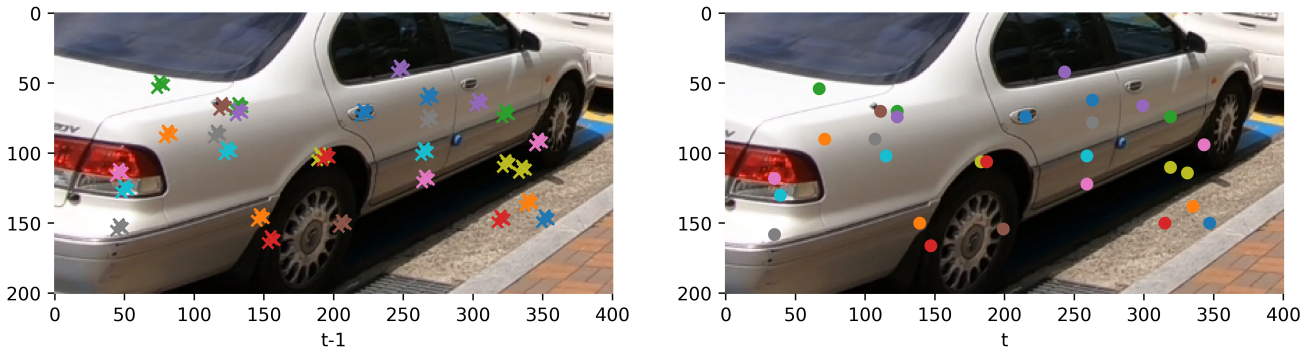

Figure 1: Illustration of offset predictions. A subset of key/value offset locations in frame x_{t-1} are shown on the left. The corresponding pixel-dense query locations in frame x_t are marked in the same colors on the right. For detailed visual inspection, the offsets are illustrated in the high-resolution domain.

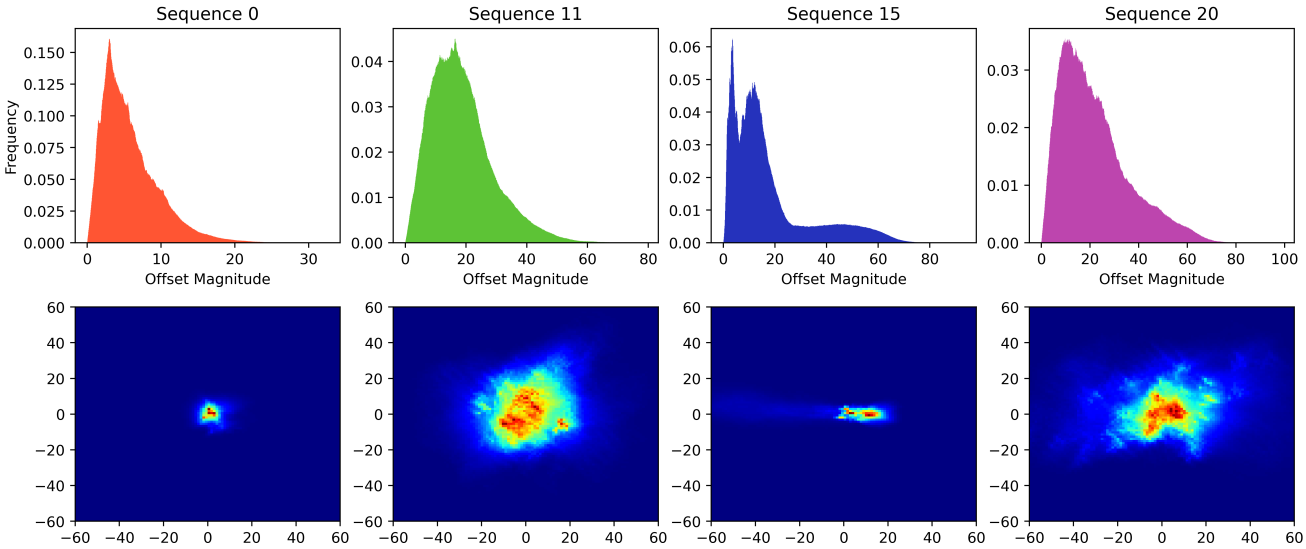

Figure 2: Analysis of offset locations for DAP-128. Histograms of offset magnitudes are plotted for each sequence in REDS (test set). The bottom row shows corresponding 2D histograms to assess the prominent orientations. Offsets are computed relative to the current frame x_t and are reported in high-resolution domain (in pixels).

REDS test set, only the first row reveals better results for forward propagation. The better performing methods include forward camera motion, while the camera in the sequence in the first row pans from left to right. In effect, the sign is first visible in higher resolution (larger) in forward evaluation, leading to better results with the same argument. The lighter model DAP-64 (reverse) even surpasses the visual quality of DAP-128 in row 4. The tiger's reconstruction shows sharper lines and reveals more details.

Configuration	$\overrightarrow{\text{DAP-64}}$	$\overleftarrow{\text{DAP-64}}$	$\overrightarrow{\text{DAP-128}}$	$\overleftarrow{\text{DAP-128}}$
REDS [11]	29.97/0.8571	**30.16/0.8635**	30.49/0.8676	**30.72/0.8751**

Table 1: Forward/Reverse ($\rightarrow$/$\leftarrow$) evaluation on REDS4 test set. We evaluate the same model in both directions.

5. Additional Vimeo-90K Results

We already report full results on REDS and UDM10 – the most relevant datasets due to their high resolution and long sequences – in the main paper along with results for Vimeo-90K with the blur/downsample kernel (BD), which provide a comprehensive overall picture of the compared methods' performance.

For completeness we additionally computed results on Vimeo-90K, obtained by application of Matlab's Bicubic downsampling kernel (BI), see Tab. 2. We selected the BD setting in the paper as more methods report their results in this setting on Vimeo-90K. The relative performance to the other methods with BI is similar to the BD setting - as generally is the case for different kernels. Thus, the discussion

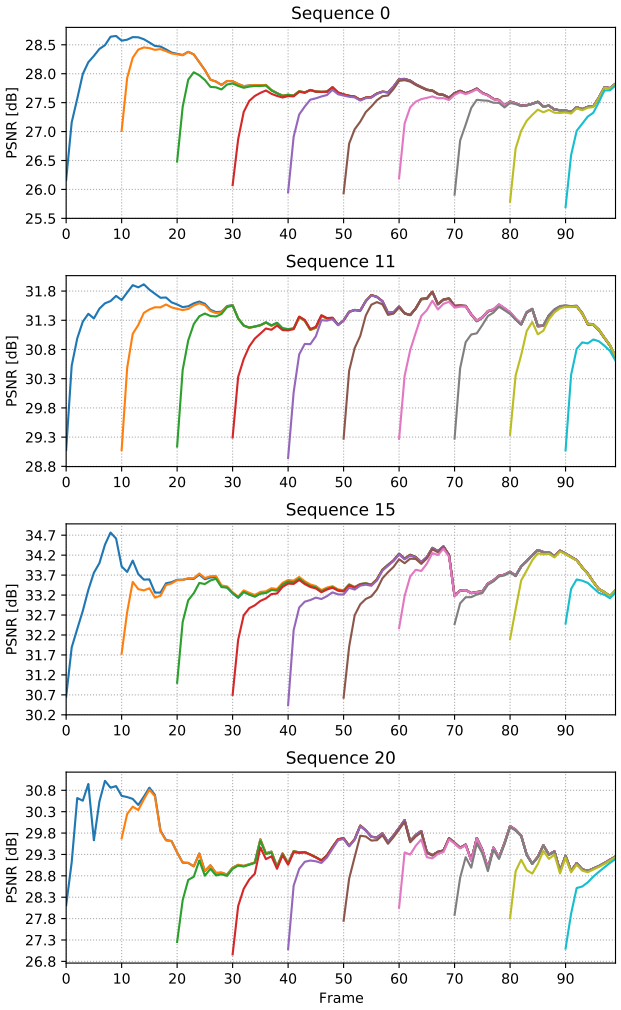

Figure 3: Analysis of information propagation in our recurrent cell for DAP-128 on REDS (test set).

and conclusions in the paper are equally valid after inspection of the BI results. Note, as explained in the paper in Sec. 4.2, Vimeo-90K has limitations due to short sequences and its evaluation protocol, which is intended for window-based methods.

6. Method Details

In this section we present additional details of our modules.

Offset Prediction Block Our offset prediction network $\mathcal{C}_{\mathcal{S}}^{l}$ is composed of several light convolution layers with leaky ReLU activations. It features an expansive part followed by a contracting part, all kernels are of size 7×7. We denote the layers as $\{f_{in}, f_{out}\}$, f_{in} represents number of input features, f_{out} stands for number of output features. Network $\mathcal{C}_{\mathcal{S}}^{l}$ is a sequence of layers in following configuration: ($\{24, 32\}$, $\{32,64\}$,$\{64,32\}$, $\{32,16\}$, $\{16,8\}$). The input consists of $8 + 8 + 4 \times 2$ features, 2 input feature maps f_t^l, v_t^l (encoded features + attention aggregated features) plus the upscaled sampling estimates $U_{\uparrow}(s_t^{l+1})$ from the previous level (k locations).

Deformable Attention Our proposed attention mechanism consists of 3 processing steps; (1) sampling, (2) encoding and (3) attention. (1) For each pixel we sample at k locations in f_{t-1}^l according to $s^l \in \mathbb{R}^{H/2^l \times W/2^l \times 2k}$, to obtain k shifted feature vectors. Note, we use 4 groups to further reduce computations. Each sampled feature vector is encoded into key/value-pairs in step (2), the pixel-dense current frame features f_t^l are encoded into query vectors. The feature size for query/key is set to 8. Then, cross-attention (3) is performed to aggregate values according to query/key correlations. In a final step, the hidden state features are aggregated. To accommodate the larger feature size in the hidden state, we encode its query/key vectors into features of size 8, but retain the feature size for the values by encoding them in their native dimension (32 per group in DAP-128). This ensures propagation of information in the hidden state without a bottle neck.

Main Processing Block The main processing block $\mathcal{N}$ consists of a convolutional layer to aggregate hidden state and input frame x_t, followed by 5 repeated fully convolutional IMDN blocks [5]. In order to produce the next hidden state h_t and the output y_t we employ another convolutional layer at the end. The input feature dimensions are set to $128 + 3$ corresponding to the feature size in the hidden state and number of color channels in x_t respectively. Following the repeated IMDN blocks, the final convolution layer produces the high-resolution output y_t, represented in low resolution (48 features) and the next hidden state h_t (128 features for DAP-128). The high-resolution output frame in RGB is obtained with pixel-shuffle. We also adopt residual learning (nearest neighbor interpolation).

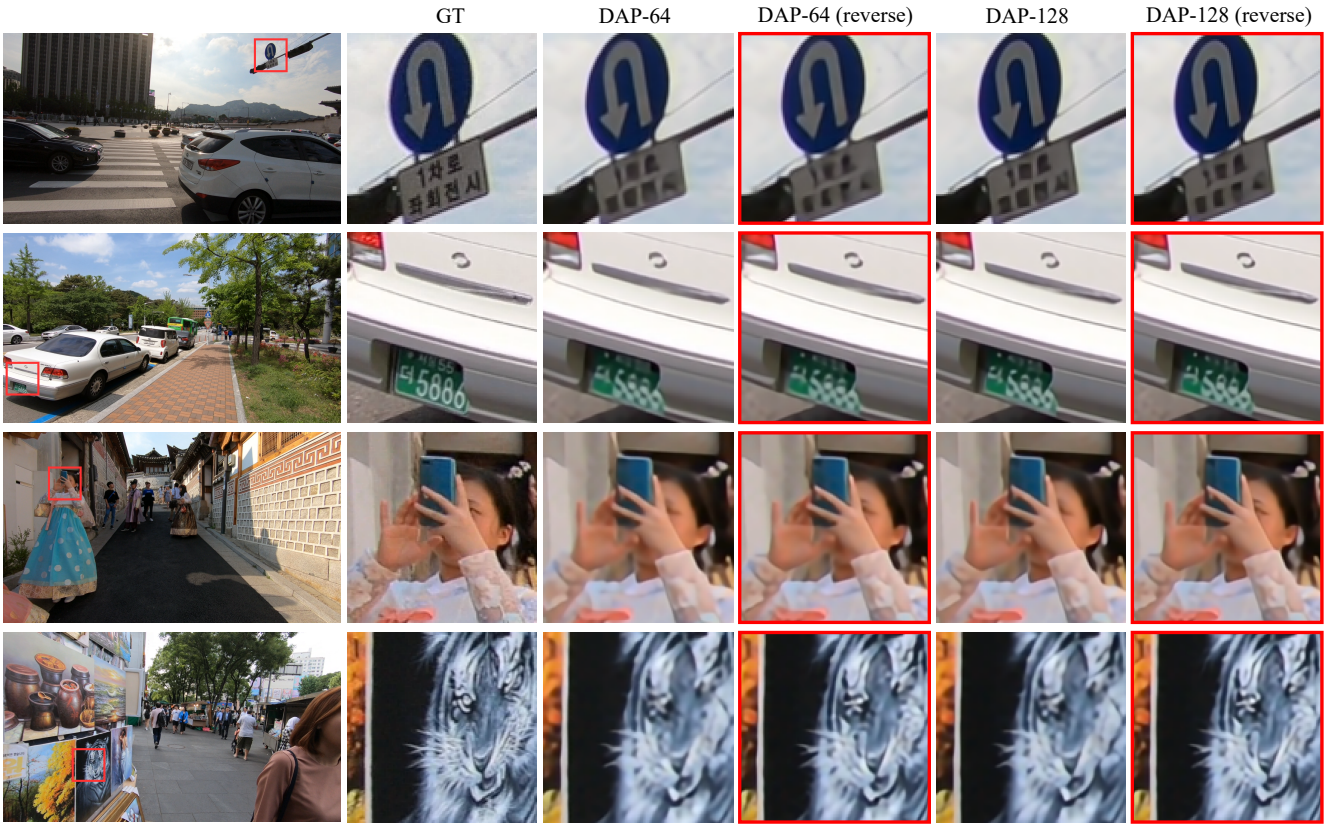

Figure 4: Visual examples on REDS (test set) for forward and reverse mode evaluation. Except for the top row, where the camera motion exhibits opposite behavior, all sequences are better reconstructed in reverse mode. Reverse mode results are highlighted with red borders.

Method	Unid.	Onl.	R-T.	Run [ms]	fps [1/s]	FLOPs [G]	MACs [G]	REDS4[11] PSNR/SSIM	UDM10[16] PSNR/SSIM	Vimeo-90K [14] BD PSNR/SSIM	Vimeo-90K [14] BI PSNR/SSIM
Bicubic	✓	✓	✓	-	-	-	-	26.14/0.7292	28.47/0.8253	31.30/0.8687	31.32/0.8684
TOFlow [15]	✓	✗	✗	-	-	-	-	27.98/0.7990	36.26/0.9438	34.62/0.9212	33.08/0.9054
FRVSR [12]	✓	✓	✗	*137	*7.3	-	-	-	37.09/0.9522	35.64/0.9319	-
DUF [9]	✓	✗	✗	*974	*1.0	-	-	28.63/0.8251	38.48/0.9605	36.87/0.9447	-
RBPN [4]	✓	✓	✗	*1507	*0.7	-	-	30.09/0.8590	38.66/0.9596	37.20/0.9458	37.07/0.9435
PFNL [16]	✓	✗	✗	*295	*3.4	-	-	29.63/0.8502	38.74/0.9627	-	36.14/0.9363
MuCAN [10]	✓	✗	✗	2'208	0.5	15'853.2	7'922.8	30.88/0.8750	-	-	37.32/0.9465
EDVR-M [13]	✓	✗	✗	116	8.6	925.7	462.3	30.53/0.8699	39.40/0.9663	37.33/0.9484	37.09/0.9446
EDVR [13]	✓	✗	✗	348	2.9	4'037.3	2'017.3	31.09/0.8800	39.89/0.9686	37.81/0.9523	37.61/0.9489
TGA [7]	✓	✗	✗	427	2.3	-	-	-	-	37.59/0.9516	-
RSDN [6]	✓	✓	✗	63	15.9	713.2	356.3	-	39.35/0.9653	37.23/0.9471	-
RRN [8]	✓	✓	✓	28	35.7	387.5	193.6	-	38.96/0.9644	-	-
RLSP [3]	✓	✗	✓	30	33.3	503.7	251.8	-	38.48/0.9606	36.49/0.9403	-
DAP-128 (ours)	✓	✓	✓	38	26.3	330.0	164.8	30.59/0.8703	39.50/0.9664	37.29/0.9476	37.06/0.9439
BasicVSR [1]	✗	✗	✗	82	12.2	754.3	376.7	31.42/0.8909	39.96/0.9694	37.53/0.9498	37.18/0.9450
IconVSR [1]	✗	✗	✗	100	10.0	904.9	451.9	31.67/0.8948	40.03/0.9694	37.84/0.9524	37.47/0.9476
BasicVSR++ [2]	✗	✗	✗	110	9.1	837.1	418.1	32.39/0.9069	40.72/0.9722	38.21/0.9550	37.79/0.9500

Table 2: Additional results with Matlab's Bicubic downsampling kernel (BI) on Vimeo-90K. Red denotes best, blue denotes second best.

References

[1] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. *arXiv preprint arXiv:2012.02181*, 2020.

[2] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. *arXiv preprint arXiv:2104.13371*, 2021.

[3] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCV Workshops*, 2019.

[4] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia (ACM MM)*, pages 2024–2032, 2019.

[6] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 645–660, Cham, 2020. Springer International Publishing.

[7] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] Takashi Isobe, Fang Zhu, and Shengjin Wang. Revisiting temporal modeling for video super-resolution. In *BMVC*, 2020.

[9] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 335–351, Cham, 2020. Springer International Publishing.

[11] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[12] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-Recurrent Video Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[13] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[14] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

[15] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

[16] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 3106–3115, 2019.