Supplementary material: Contrastive Learning for Controllable Blind Video Restoration

Givi Meishvili[1]
gmeishvili@microsoft.com
Abdelaziz Djelouah[2]
abdelaziz.djelouah@disney.com
Sally Hattori[3]
sally.hattori@disney.com
Christopher Schroers[2]
christopher.schroers@disney.com

[1] Microsoft
(Work was done at DisneyResearch|Studios, before joining Microsoft)

[2] DisneyResearch|Studios
Zurich, Switzerland

[3] The Walt Disney Company
Los Angles, USA

1 Implementation & Training Details

Our method takes five consecutive distorted frames as input and produces a restored middle frame. We employed R3D-18 [8] architecture as a backbone for the encoder E_d. Before contrasting, the output features of encoder E_d are fed to two fully connected layers with 512 and 256 neurons, respectively. R_b, R_{SR}, and R_{DN} models consist of Degradation-aware blocks introduced by Wang *et al*. [10]. Models E_s, E_k, and M consist of 2, 2 and 3 fully connected layers, respectively.

Regarding degradations, during training, we sample anisotropic Gaussian blur kernels and additive white gaussian noise. We train our models on the Vimeo90K dataset[11]. This dataset consists of 89,800 video clips, which cover a large variety of scenes and actions. During training, we randomly sample 5 consecutive 192x192 frames from the dataset. Our models were trained on 2 NVIDIA TITAN X GPUs with a mini-batch size of 32 samples. In our experiments, we used $\tau = 0.07$ and $N_Q = 8192$, respectively. Degradation encoder E_d was pre-trained for 65 epochs prior to the final fine-tuning together with models R_b, R_{SR}, and R_{DN} for additional 40 epochs. We use the Adam optimizer [9] with an initial learning rate of 10^{-4} to train all the networks.
Hyper-parameter values for our losses are set as follows: $\lambda_{SR} = 1$, $\lambda_{DN} = 1$, $\lambda_c = 1$, $\lambda_k = 400$, and $\lambda_s = 1$.

2 Handling Temporal Information

Handling temporal information is one of the crucial points arising while designing video restoration methods. Several works in the field first explicitly incorporate optical flow or

motion estimation step and after perform feature warping and fusion [4, 5]. Others design restoration architectures that result in inductive biases that encourage the effective use of temporal information [6, 7]. In our work, temporal information is handled by processing the distorted input frames using a 3D convolutional encoder E_d based on R3D-18 residual architecture, which is effective for processing videos [8]. We also employ several 3D convolutional layers in our restoration backbone R_B. Even though we don't use any implicit alignment or motion compensation step, the quantitative comparisons in the main paper show that our method outperforms video super-resolution and denoising methods on conventional video test sets.

3 Degradation Encoders & Mutator

This section shows some architectural details of the E_d, MLP head, E_k, E_s, and M models.

Encoder E_d. We employed R3D-18 [8] architecture as a backbone for the encoder E_d. Contrastive MLP head consists of the layers presented in Table 1.

Encoder E_k, E_s. Encoders E_k and E_s are implemented using fully-connected perceptrons, presented in Tables 2 and 3, respectively.

Degradation Mutator M. Degradation mutator M is implemented using 3 fully-connected layers, presented in Table 4.

4 Restoration Models

This section shows some architectural details of the R_B, R_{SR}, and R_{DN} models. The main building block of our models is the Degradation-Aware (DA) restoration block introduced in [10]. The high-level structure is borrowed from the RCAN [12] model.

Restoration Backbone R_B. Our model R_B starts with three 3d convolutional layers to leverage the temporal information presented in the input. Next, the aggregated feature is processed using 5 DA [10] blocks. Finally, we give the output of the last block to models R_{SR} and R_{DN} as input.

Super-Resolution Branch R_{SR}. We pass the input feature map from model R_B to 2 consecutive DA blocks[10]. The output of the last DA block is summed with the middle frame feature of the first convolutional layer of model R_B. Finally, we get the super-resolved output using the commonly used Pixel-Shuffle layer [9]. Alternatively, instead of Pixel-Shuffle, we employ Meta Upscale module [1] at the end of our R_{SR} model to enable non-integer upsampling factors and address more general scenarios.

Denoising Branch R_{DN}. We pass the input feature map from model R_B to 2 consecutive DA blocks[10]. The output of the last DA block is summed with the middle frame feature of the first convolutional layer of model R_B. Finally, we get the denoised output using the last conv_2d layer.

MLP **Head of Degradation Encoder** E_d					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	1024	-	-	`lReLU`	512
dense	512	-	-	`linear`	256

Table 1: The network architecture of contrastive *MLP* head. The input pairwise concatenated features are of size 1024. The output size is 256.

Encoder E_k					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	512	-	-	`lReLU`	441
dense	441	-	-	`lReLU`	441
dense	441	-	-	`Softmax`	441

Table 2: The network architecture of encoder E_k. The input degradation embedding is a 512-dimensional vector. The output size is 441, giving 21×21 blur kernel after reshaping.

Encoder E_s					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	512	-	-	`lReLU`	128
dense	128	-	-	`linear`	1

Table 3: The network architecture of encoder E_k. The input degradation embedding is a 512-dimensional vector. The output size is 1.

Degradation Mutator M					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	954	-	-	`lReLU`	954
dense	954	-	-	`lReLU`	954
dense	954	-	-	`linear`	512

Table 4: The network architecture of mutator M. The input degradation embedding is 954 dimensional vector (since feature from E_d is 512 dimensional, reshaped input kernel is $21 \times 21 = 441$ dimensional and the input noise level is 1 dimensional). The output size is a 512-dimensional vector that matches the output size of E_d.

Restoration Backbone R_B					
Layer	Kernel	Stride	Norm.	Activation	# Filters
conv_3d	$1 \times 5 \times 5$	1	-	`linear`	128
conv_3d	$3 \times 7 \times 7$	1	-	`lReLU`	128
conv_3d	$3 \times 7 \times 7$	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128

Table 5: The network architecture of restoration backbone R_B.

Super-Resolution Branch R_{SR}					
Layer	Kernel	Stride	Norm.	Activation	# Filters
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
Pixel-Shuffle	-	1	-	`linear`	3

Table 6: The network architecture of super-resolution branch R_{SR}.

Denoising Branch R_{DN}					
Layer	Kernel	Stride	Norm.	Activation	# Filters
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
conv_2d	3×3	1	-	`linear`	3

Table 7: The network architecture of denoising branch R_{DN}.

5 Qualitative Results

This section shows qualitative results of the best-performing methods from Tables 2, 3, and 4 of the main paper. We performed a qualitative comparison with some of the state-of-the-art super-resolution [3, 7], denoising [6], and scratch removal [9] methods.

Video Super-Resolution. In Figure 1, we performed a qualitative comparison in video super-resolution with Tian *et al*. [7] and Pan *et al*. [3].

Video Denoising. We performed a qualitative comparison with the video denoising method of Tassano *et al*. [6]. Results are presented in Figure 2.

Video Scratch Removal. We performed a qualitative comparison with the scratch removal method of Wan *et al*. [9]. Results are presented in Figure 3.

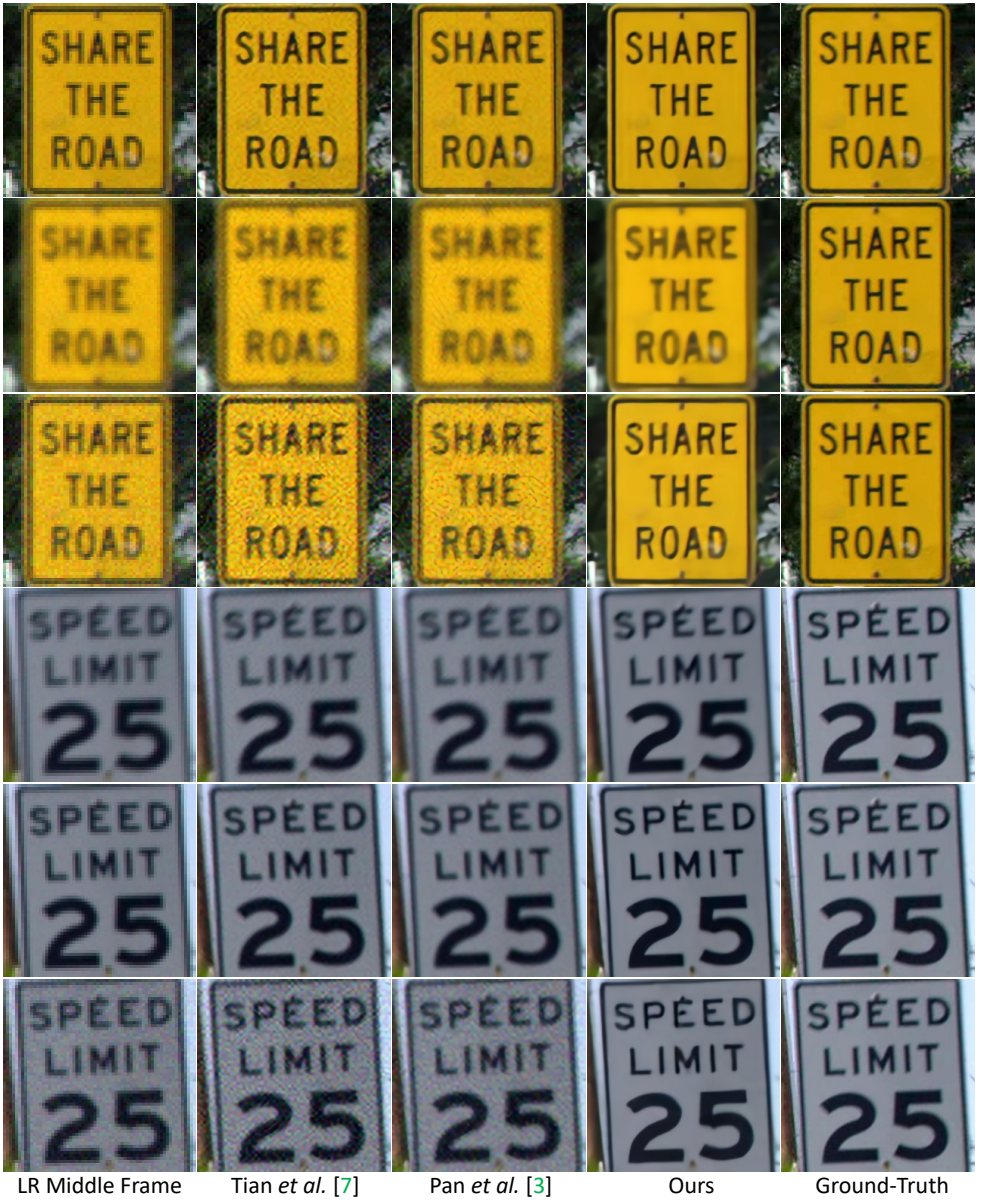

Figure 1: **Qualitative Comparison Super-Resolution.** We performed a qualitative comparison with methods of Tian *et al*. [7] and Pan *et al*. [3]. Different rows correspond to different combinations of blur kernels and noise levels. The first column corresponds to a low-resolution input middle frame. Next, the second and third columns correspond to the restored results of Tian *et al*. [7] and Pan *et al*. [3], respectively. The fourth column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

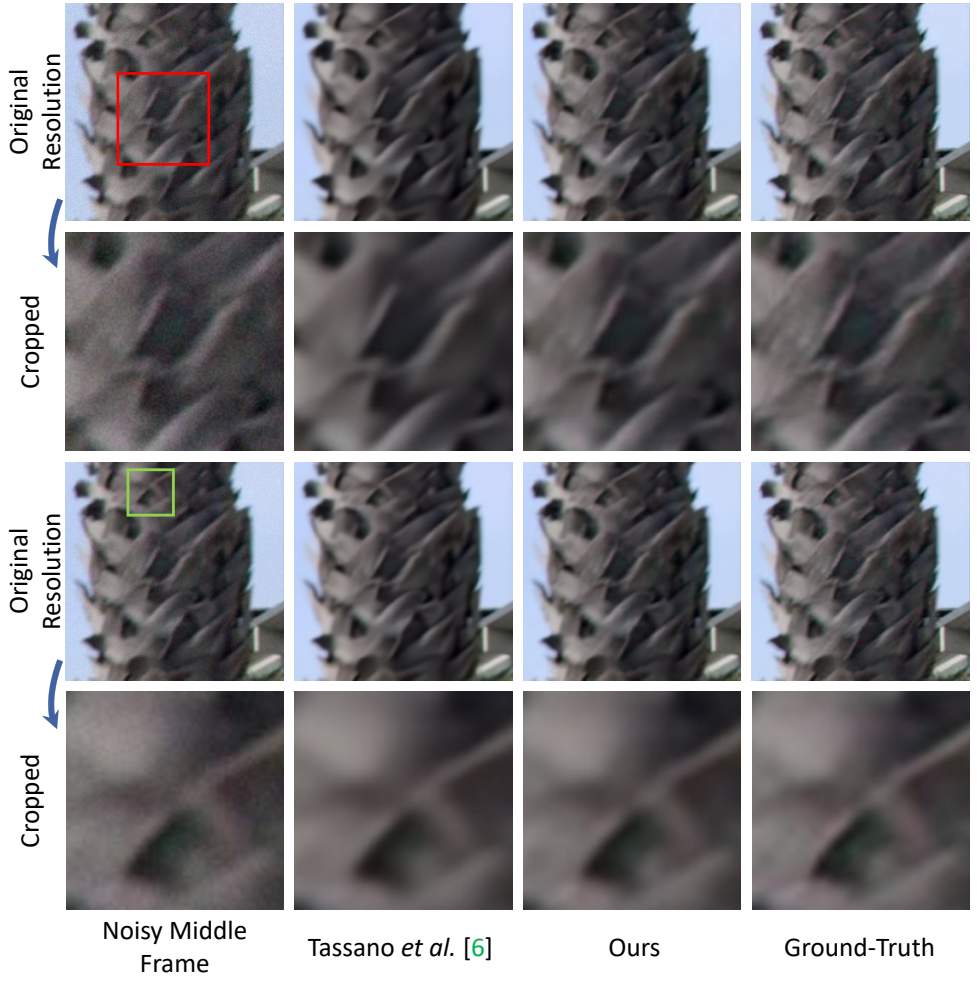

Figure 2: **Qualitative Comparison Denoising.** We performed a qualitative comparison with the method of Tassano *et al*. [6]. The first two rows correspond to a noise level of $\sigma = 25$. The last two rows correspond to a noise level of $\sigma = 15$. The first column corresponds to a noisy input middle frame. The second column corresponds to the restored results of Tassano *et al*. [6]. The third column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

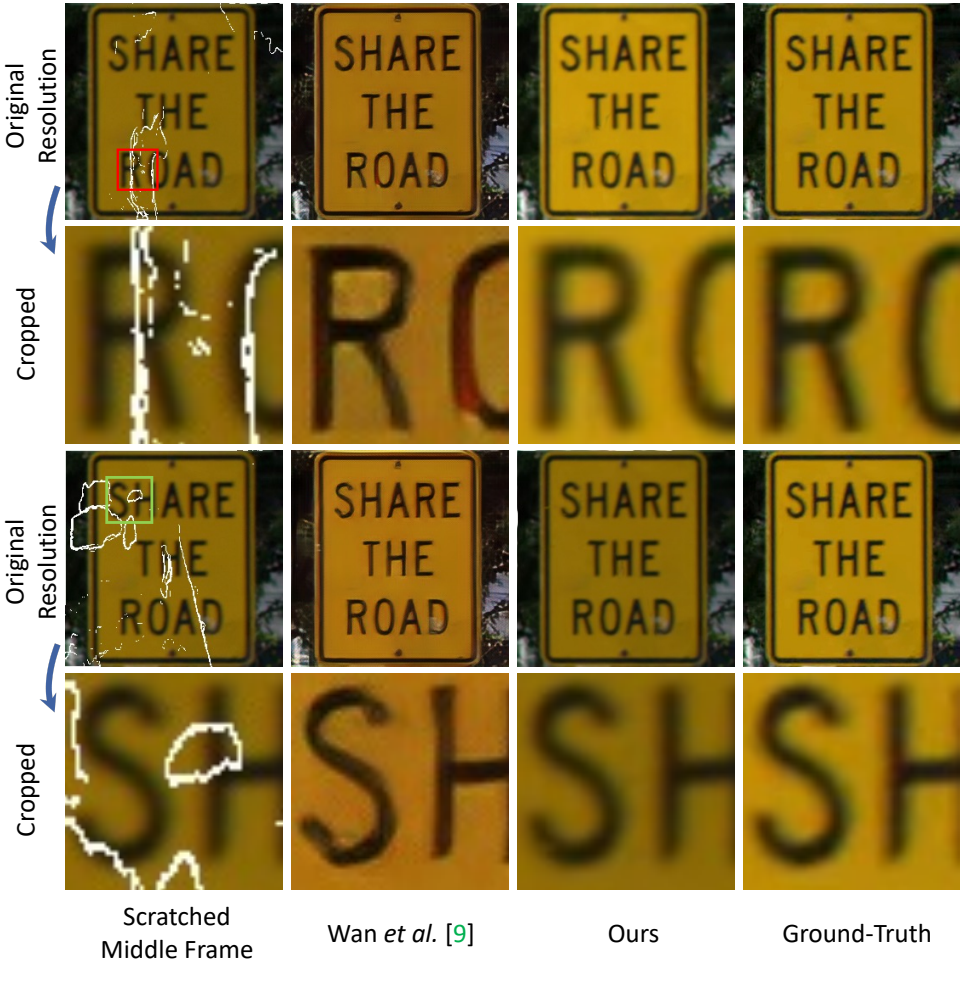

Figure 3: **Qualitative Comparison Scratch Removal.** We performed a qualitative comparison with the method of Wan *et al*. [9]. The first column corresponds to a scratched input middle frame. The second column corresponds to the restored results of Wan *et al*. [9]. The third column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

References

[1] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. 2019.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Jinshan Pan, Haoran Bai, Jiangxin Dong, Jiawei Zhang, and Jinhui Tang. Deep blind video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4811–4820, October 2021.

[4] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1805–1809. IEEE, 2019.

[6] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, and Fang Wen. Bringing old photos back to life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2747–2757, 2020.

[10] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10581–10590, June 2021.

[11] 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.

[12] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.