Dual-Stream Fusion Network for Spatiotemporal Video Super-Resolution
Supplementary Materials

Min-Yuan Tseng[†] Yen-Chung Chen[†] Yi-Lun Lee[†]
Wei-Sheng Lai[‡] Yi-Hsuan Tsai[§] Wei-Chen Chiu[†]
[†]National Chiao Tung University [‡]Google [§]NEC Labs America

1. Introduction

In this supplementary document, we provide additional discussions, details, and results to complement the paper. First, we provide more discussion on investigating different model variants. Second, we provide the detailed model architecture of the fusion and refinement networks. Third, we show additional visual results for qualitative comparisons.

2. More Investigation and Discussion

Baselines of using video super-resolution method as spatial super-resolution module. We include two baselines of using RBPN [1] (in which we use the pretrained model provided by the authors) as the spatial upsampling sub-network, with the quantitative results shown in Table 1. We can see that these two new baselines have inferior performance with respect to the ones of using the image super-resolution methods as our spatial upsampling module.

Table 1: Quantitative comparisons among different combinations of upsampling sub-networks.

Vimeo-90K	$\hat{H}^{(t)}_{T\rightarrow S}$	
	PSNR	SSIM
RBPN + SuperSloMo	29.20	0.8984
RBPN + DAIN	28.74	0.8929
ESPCN + SuperSloMo	31.41	0.9179
ESPCN + DAIN	31.67	0.9248
SAN + SuperSloMo	31.73	0.9225
SAN + DAIN	31.93	0.9279

Baselines of having refinement network on single stream. In our proposed framework there is no refinement network attached on either the $\mathbb{M}_{T\rightarrow S}$ or the $\mathbb{M}_{S\rightarrow T}$ stream. Here we build the model variants with adding refinement network $\mathbb{R}$ after $\mathbb{M}_{T\rightarrow S}$ or $\mathbb{M}_{S\rightarrow T}$, and show the comparison as Table 2 (where $\mathbb{M}_S$ and $\mathbb{M}_T$ are ESPCN and SuperSloMo respectively). We can see that even when we have the additional refinement network attached onto the single stream, the resultant performance is still inferior to the fusion over two streams, thus verifying again the contribution of our dual-stream fusion framework.

3. Model Architectures

1) Fusion network $\mathbb{F}$. Table 3 shows the detailed architecture of our fusion network $\mathbb{F}$, which is a U-Net architecture with five symmetric downsampling and upsampling convolutional blocks. Each convolutional layer is followed by the leaky ReLU activation except the last layer, which uses the sigmoid function to ensure the output masks are in $[0, 1]$.

2) Refinement network $\mathbb{R}$. As shown in Table 4, the refinement network $\mathbb{R}$ is composed of 8 convolutional layers without any downsampling or upsampling operations.

Table 2: Quantitative comparisons with the model variants of adding refinement network onto the single stream.

Setting	Vimeo-90K		UCF101	
	PSNR	SSIM	PSNR	SSIM
$\mathbb{M}_{T\to S}$ (fixed) + $\mathbb{R}$	31.94	0.9285	30.98	0.9274
$\mathbb{M}_{S\to T}$ (fixed) + $\mathbb{R}$	31.87	0.9287	31.21	0.9298
$\mathbb{M}_{T\to S}$ (fixed) + $\mathbb{M}_{S\to T}$ (fixed) + $\mathbb{F}$	32.23	0.9313	31.38	0.9308
$\mathbb{M}_{T\to S} + \mathbb{M}_{S\to T} + \mathbb{F} + \mathbb{R}$ (jointly fine-tuned)	**32.85**	**0.9401**	**31.54**	**0.9317**

Table 3: **Architecture of fusion network** $\mathbb{F}$

Layer	Input Channels	Output Channels	Kernel Size	Activation
conv1_1	3	32	7 × 7	-
conv1_2	32	32	7 × 7	-
Average Pooling 2×				
conv2_1	32	64	5 × 5	Leaky ReLU
conv2_2	64	64	5 × 5	Leaky ReLU
Average Pooling 2×				
conv3_1	64	128	3 × 3	Leaky ReLU
conv3_2	128	128	3 × 3	Leaky ReLU
Average Pooling 2×				
conv4_1	128	256	3 × 3	Leaky ReLU
conv4_2	256	256	3 × 3	Leaky ReLU
Average Pooling 2×				
conv5_1	256	512	3 × 3	Leaky ReLU
conv5_2	512	512	3 × 3	Leaky ReLU
Average Pooling 2×				
conv6_1	512	512	3 × 3	Leaky ReLU
conv6_2	512	512	3 × 3	Leaky ReLU
Bilinear Upsampling 2×				
conv7_1	512	512	3 × 3	Leaky ReLU
conv7_2	512+512	512	3 × 3	Leaky ReLU
Bilinear Upsampling 2×				
conv8_1	512	256	3 × 3	Leaky ReLU
conv8_2	256+256	256	3 × 3	Leaky ReLU
Bilinear Upsampling 2×				
conv9_1	256	128	3 × 3	Leaky ReLU
conv9_2	128+128	128	3 × 3	Leaky ReLU
Bilinear Upsampling 2×				
conv10_1	128	64	3 × 3	Leaky ReLU
conv10_2	64+64	64	3 × 3	Leaky ReLU
Bilinear Upsampling 2×				
conv11_1	64	32	3 × 3	Leaky ReLU
conv11_2	32+32	32	3 × 3	Leaky ReLU
conv12_1	32	3	3 × 3	Sigmoid

4. Additional Visual Comparisons

1) Additional comparisons with FISR. We show more qualitative comparisons between the proposed method and FISR [15] on the FISR test set in Figure 1, and on the Vimeo-90K test set in Figure 2 and Figure 3, respectively.

2) Additional visual results. In Figure 4 and 5, we show more visual comparisons between the proposed method and its variants. Moreover, we provide several video clips for comparisons in the supplementary video. Overall, the proposed

Table 4: **Architecture of refinement network** $\mathbb{R}$

Layer	Input Channels	Output Channels	Kernel Size	Activation
conv1_1	3	128	7 × 7	ReLU
conv2_1	128	128	3 × 3	ReLU
conv2_2	128	128	3 × 3	ReLU
conv3_1	128	128	3 × 3	ReLU
conv3_2	128	128	3 × 3	ReLU
conv4_1	128	128	3 × 3	ReLU
conv4_2	128	128	3 × 3	ReLU
conv5_1	128	3	3 × 3	-

method is able to generate finer details and better reconstructions.

References

[1] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

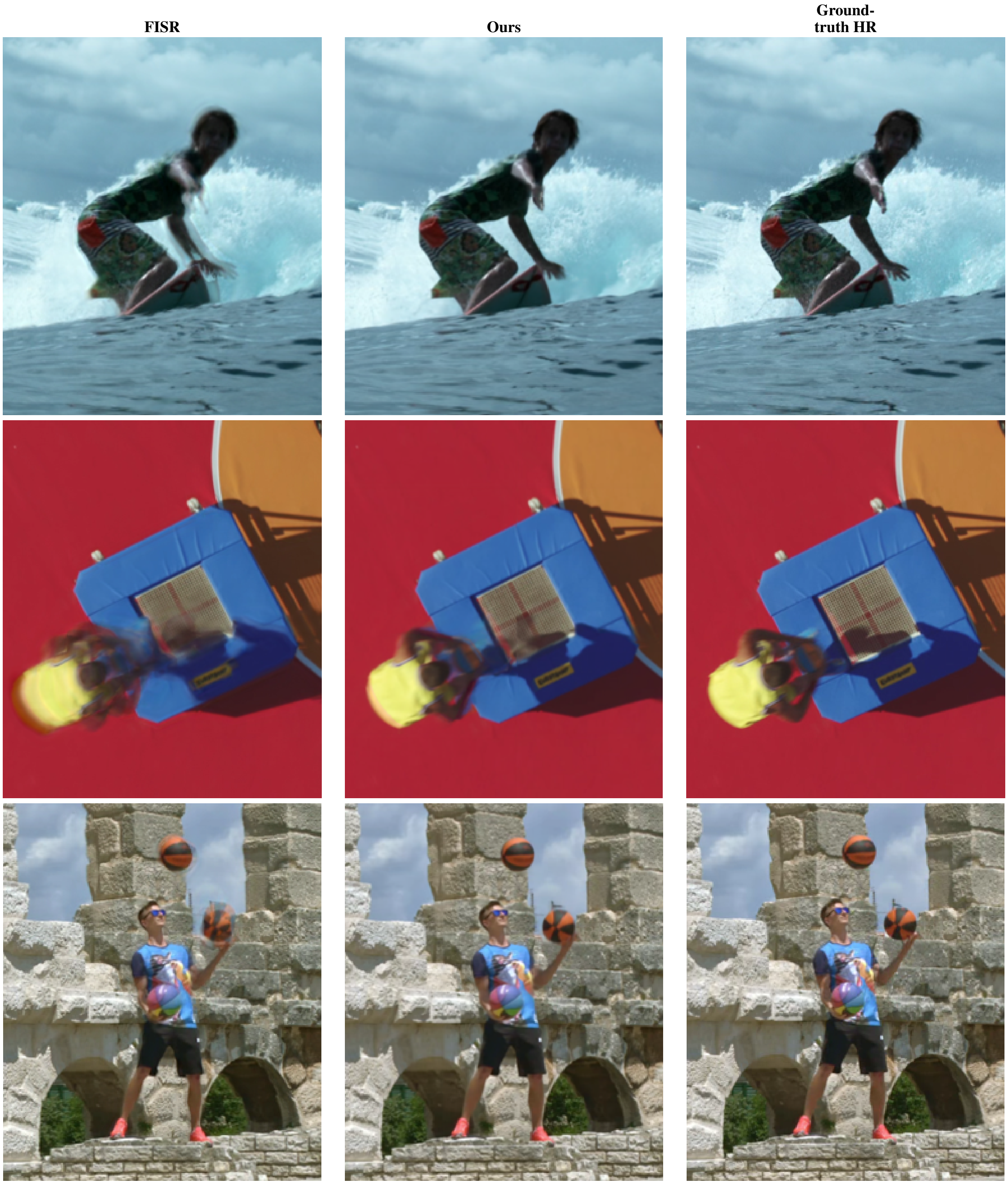

Figure 1: **Qualitative comparison with the state-of-the-art spatiotemporal upsampling method, FISR.** Our method is able to produce the clearer upsampling results with fewer artifacts in comparison to FISR.

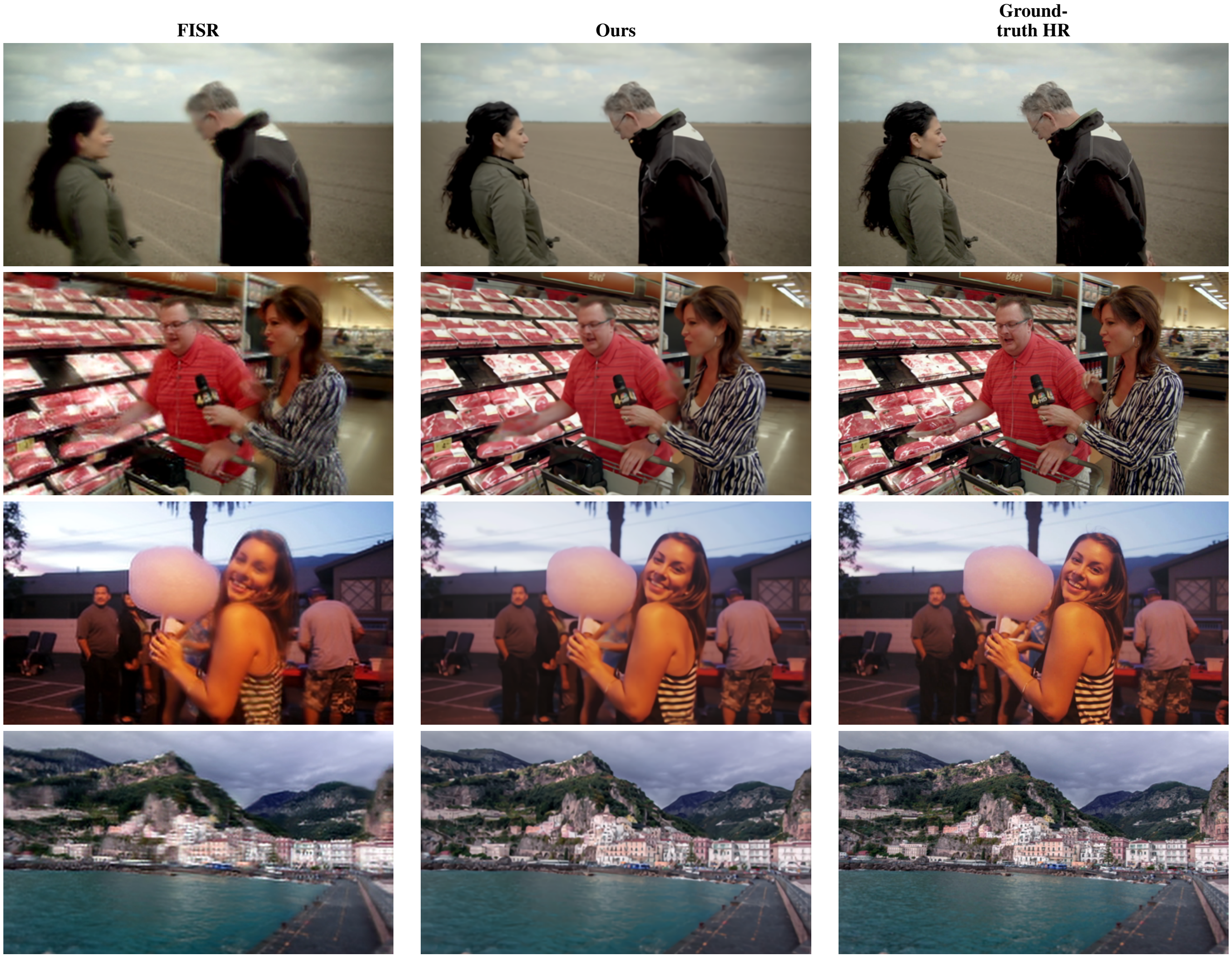

Figure 2: **Additional visual comparison with the state-of-the-art method, FISR.** Our proposed method is able to produce the spatiotemporally upsampled results with fewer artifacts.

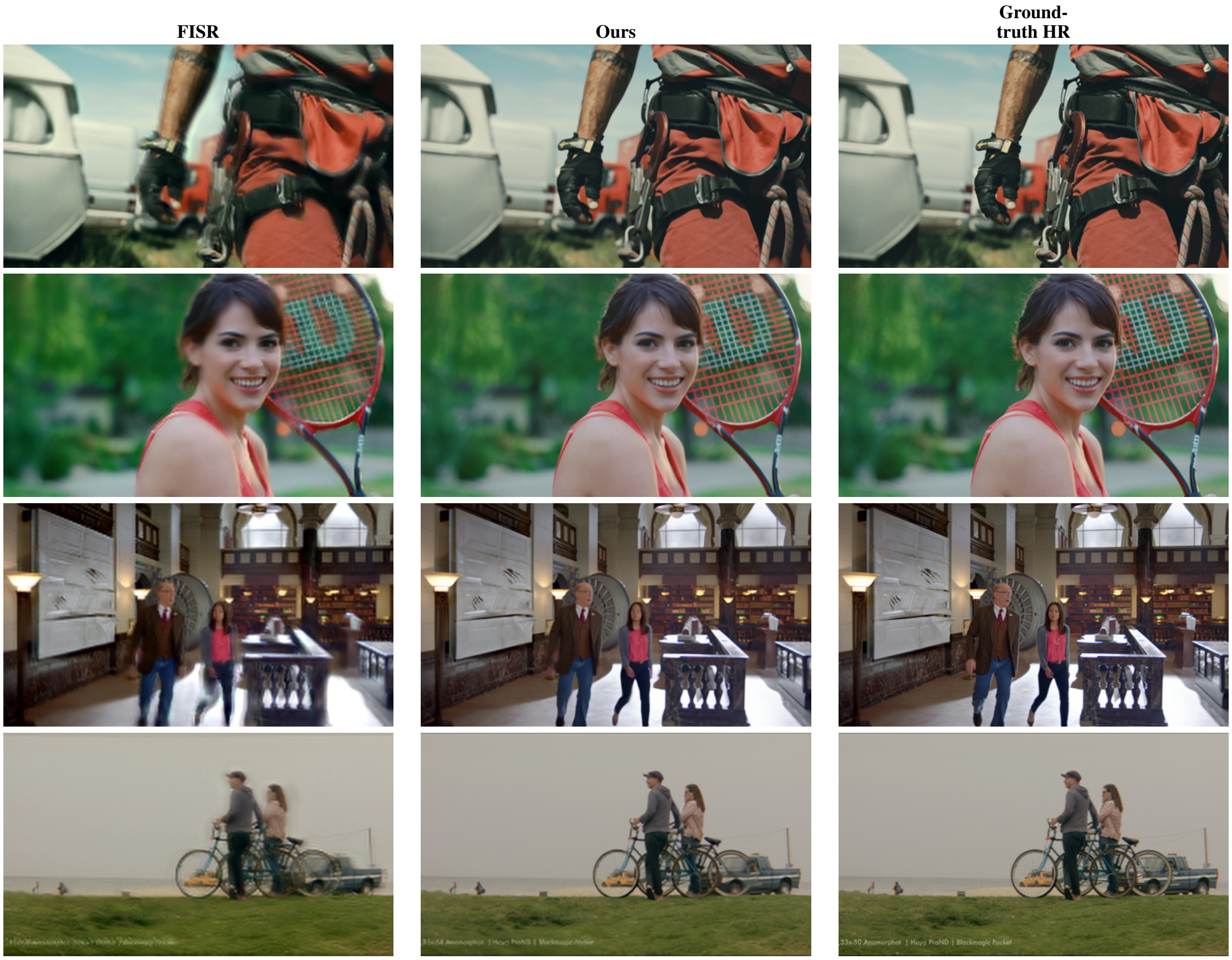

Figure 3: **Additional visual comparison with the state-of-the-art method, FISR.** Our proposed method is able to produce the spatiotemporally upsampled results with fewer artifacts.

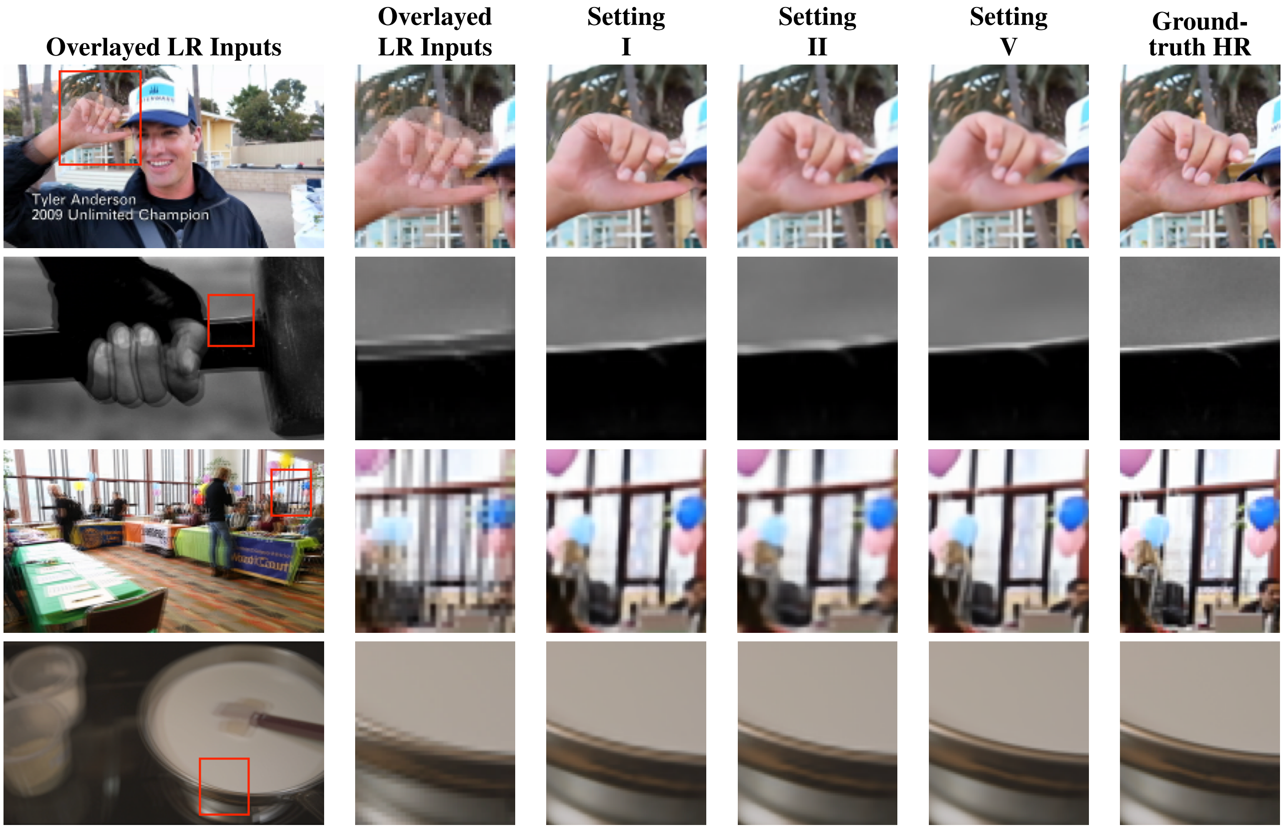

Figure 4: **Visual comparisons between the results from different training stages of the proposed framework.** Please refer to Table 2 of the main paper for the specific setting of each variant.

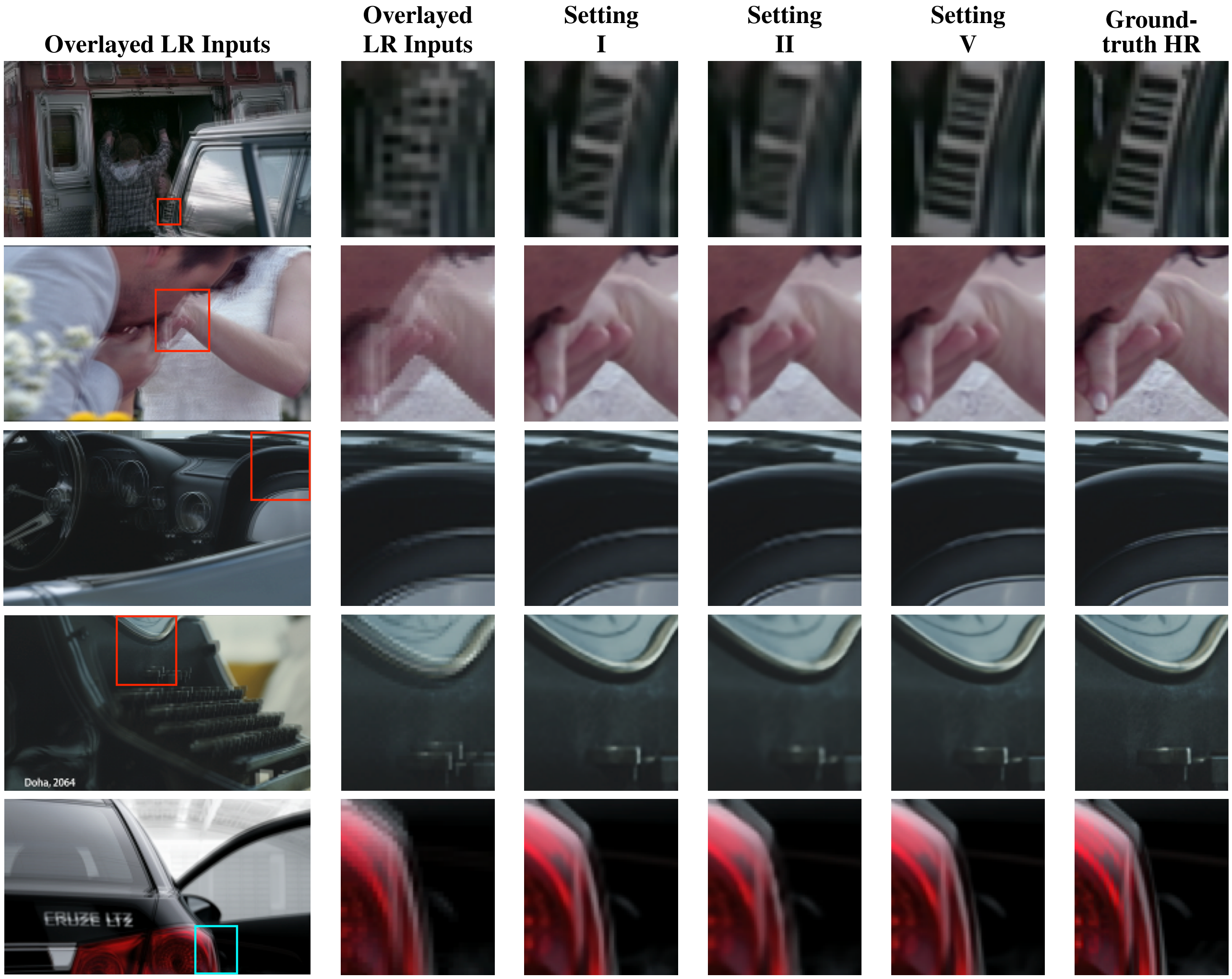

Figure 5: **Visual comparisons between the results from different training stages of the proposed framework.** Please refer to Table 2 of the main paper for the specific setting of each variant.