Joint Video Multi-Frame Interpolation and Deblurring under Unknown Exposure Time
Supplementary Material

Wei Shang[1], Dongwei Ren[1*], Yi Yang[1], Hongzhi Zhang[1], Kede Ma[2], Wangmeng Zuo[1,3]
[1]School of Computer Science and Technology, Harbin Institute of Technology
[2] City University of Hong Kong [3]Peng Cheng Laboratory, Shenzhen

In this supplementary file, we provide details of the network architecture of the proposed VIDUE method and more results on benchmark datasets. Please refer to the link for more visual results at ***https://onedrive.live.com***.

1. Network Architecture

As mentioned in the main manuscript, the proposed VIDUE consists of an exposure-aware feature extractor g_e, an intra- and inter-motion analyzer g_a, and a video reconstruction network f.

1.1. Exposure-Aware Feature Extractor

We illustrate the network structure of the exposure-aware feature extractor g_e in Table S1, in which ResBlock is detailed in Table S2.

1.2. Intra- and Inter-Motion Analyzer

We choose to analyze both intra-motion within each video frame and inter-motion between frames. Our motion analyzer $g_a : \mathbb{R}^{(T\times 3)\times H\times W} \mapsto \mathbb{R}^{T\times 1\times H\times W} \times \mathbb{R}^{(S\times T)\times 2\times H\times W}$, computes, from an input video sequence, T intra-motion maps and $(S\times T)\times 2$ inter-motion maps of the same size, respectively. We show the network structure for intra-motion analysis in Table S3. The network structure for inter-motion analysis is similar to RefineNet proposed in [3], which is detailed in Table S4.

1.3. Video Reconstruction Network

The video reconstruction network $f : \mathbb{R}^{T\times 3\times H\times W} \mapsto \mathbb{R}^{(S\times T)\times 3\times H\times W}$ is built upon the exposure-aware representation $\boldsymbol{u}$ and motion-aware representations $\boldsymbol{m}$ and $\boldsymbol{n}$ by progressive exposure-adaptive convolution and motion refinement. The network structure is illustrated in Table S5.

*Corresponding author: rendongweihit@gmail.com.

2. More Results on Benchmark Datasets

In this section, we provide more visual results of VIDUE in comparison to existing video interpolation and deblurring methods.

2.1. More Results on the GoPro Dataset

On the GoPro dataset with mild blur, VIDUE generally achieves more visually appealing results with sharper boundaries and fewer artifacts, as shown in Figs. S1 and S2.

2.2. More Results on the Adobe Dataset

On the Adobe dataset with heavy blur, VIDUE still performs favorably as shown in Figs. S3 and S4.

2.3. More Results on the RealBlur Dataset

On the RealBlur dataset, VIDUE generates the least artifacts and sharper textures compared with other methods, as shown in Figs. S5 and S6.

2.4. More Results on Video $\times 16$ Interpolation and Deblurring

Although video $\times 16$ interpolation and deblurring is extremely difficult (without the help of extra auxiliary information), VIDUE is better at it than others, as shown in Figs. S7 and S8.

References

[1] Seungjun Nah, Tae H Kim, and Kyoung M Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 2, 4, 6, 7

[2] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017. 4, 5, 6

[3] Xiangyu Xu, Siyao Li, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, pages 1–10, 2019. 1

Input	A sequence of blurred frames ($\text{indim} = T \times 3$)
Layer 1	$\text{Conv}(\text{indim}, 64, 7, 1, 3)$; BN; LeakyReLU; $\text{ResBlock}(64, 64, 3, 1, 1, \text{BN} = \text{True}) \times 2$
Layer 2	$\text{Conv}(64, 128, 3, 2, 1)$; BN; LeakyReLU; $\text{ResBlock}(128, 128, 3, 1, 1, \text{BN} = \text{True}) \times 2$
Layer 3	$\text{Conv}(128, 256, 3, 2, 1)$; BN; LeakyReLU; $\text{ResBlock}(256, 256, 3, 1, 1, \text{BN} = \text{True}) \times 2$
Layer 4	$\text{Conv}(256, 256, 3, 2, 1)$; BN; LeakyReLU; $\text{ResBlock}(256, 256, 3, 1, 1, \text{BN} = \text{True}) \times 2$
Layer 5	GlobalAvgPool
Layer 6	$\text{Linear}(256, 1024)$; LeakyReLU; $\text{Linear}(1024, 256)$
Output	Exposure-aware feature representation $\boldsymbol{u}$

Table S1. The structure of the exposure-aware feature extractor g_e. T is the length of the input video sequence. BN stands for batch normalization. Convolution is in the form of (input channel, output channel, kernel size, stride, padding). The linear (*i.e.*, fully connected) layer is in the form of (input channel, output channel).

Layer 1	$\text{Conv}(D, D, 3, 1, 1)$; BN; LeakyReLU
Layer 2	$\text{Conv}(D, D, 3, 1, 1)$; BN
	Residual connection between the input and the output of *Layer 2* via addition

Table S2. The structure of ResBlock. We may remove BN by setting the corresponding indicator variable to False. D is the channel dimension of the feature maps from the previous layer.

Figure S1. More visual results on GoPro [1] ($\times 8$). Blue and red patches are from deblurred and interpolated frames, respectively. TSP is short for CDVD-TSP, and MIMO is short for MIMOUNetPlus.

Input	One blurred frame ($\text{indim} = 3$)
Layer 1	$\text{PixelUnshuffle}(2)$
Layer 2	$\text{Conv}(\text{indim} \times 4, 16, 5, 1, 2)$; $\text{ResBlock}(16, 16, 5, 1, 2, \text{BN} = \text{False}) \times 3$
Layer 3	$\text{Conv}(16, 32, 5, 2, (1, 2, 1, 2))$; $\text{ResBlock}(32, 32, 5, 1, 2, \text{BN} = \text{False}) \times 3$
Layer 4	$\text{Conv}(32, 64, 5, 2, (1, 2, 1, 2))$; $\text{ResBlock}(64, 64, 5, 1, 2, \text{BN} = \text{False}) \times 3$
Layer 5	$\text{Conv}(64, 128, 1, 1, 0)$; LeakyReLU; $\text{Conv}(128, 64, 3, 1, 1)$; LeakyReLU
Layer 6	$\text{SE}(256, 64, 4)$; $\text{TransConv}(64, 32, 4, 2, 1)$; LeakyReLU
	Concatenation of outputs from *Layer 6* and *Layer 3*
Layer 7	$\text{Conv}(64, 128, 1, 1, 0)$; LeakyReLU; $\text{Conv}(128, 64, 3, 1, 1)$; LeakyReLU
Layer 8	$\text{SE}(256, 64, 4)$; $\text{TransConv}(64, 16, 4, 2, 1)$; LeakyReLU
	Concatenation of outputs from *Layer 8* and *Layer 2*
Layer 9	$\text{Conv}(32, 64, 1, 1, 0)$; LeakyReLU; $\text{Conv}(64, 32, 3, 1, 1)$; LeakyReLU
Layer 10	$\text{SE}(256, 32, 4)$; $\text{TransConv}(32, 32, 4, 2, 1)$; LeakyReLU
Layer 11	$\text{Conv}(32, 4, 5, 1, 2)$
Output	Motion offsets

Table S3. The structure of intra-motion analysis. Adapting to $\boldsymbol{u}$ via gain tuning is in the form of squeeze-and-excitation (SE) (vec channel, input channel, reduction). Transposed convolution is in the same format as convolution.

Input	A sequence of motion offsets ($\mathrm{indim} = T \times 4$)
Layer 1	$\mathrm{Conv}(\mathrm{indim}, 32, 7, 1, 3)$; LeakyReLU; $\mathrm{Conv}(32, 32, 7, 1, 3)$; LeakyReLU
Layer 2	$\mathrm{AvgPool}(2)$; $\mathrm{Conv}(32, 64, 5, 1, 2)$; LeakyReLU; $\mathrm{Conv}(64, 64, 5, 1, 2)$; LeakyReLU
Layer 3	$\mathrm{AvgPool}(2)$; $\mathrm{Conv}(64, 128, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(128, 128, 3, 1, 1)$; LeakyReLU
Layer 4	$\mathrm{AvgPool}(2)$; $\mathrm{Conv}(128, 256, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(256, 256, 3, 1, 1)$; LeakyReLU
Layer 5	$\mathrm{AvgPool}(2)$; $\mathrm{Conv}(256, 512, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(512, 512, 3, 1, 1)$; LeakyReLU
Layer 6	$\mathrm{AvgPool}(2)$; $\mathrm{Conv}(512, 512, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(512, 512, 3, 1, 1)$; LeakyReLU
Layer 7	$\mathrm{Up}(2)$; $\mathrm{Conv}(512, 512, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(512, 512, 3, 1, 1)$; LeakyReLU; $\mathrm{SE}(256, 512, 4)$
Layer 8	$\mathrm{Up}(2)$; $\mathrm{Conv}(512, 256, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(256, 256, 3, 1, 1)$; LeakyReLU; $\mathrm{SE}(256, 256, 4)$
Layer 9	$\mathrm{Up}(2)$; $\mathrm{Conv}(256, 128, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(256, 128, 3, 1, 1)$; LeakyReLU; $\mathrm{SE}(256, 128, 4)$
Layer 10	$\mathrm{Up}(2)$; $\mathrm{Conv}(128, 64, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(64, 64, 3, 1, 1)$; LeakyReLU; $\mathrm{SE}(256, 64, 4)$
Layer 11	$\mathrm{Up}(2)$; $\mathrm{Conv}(64, 32, 3, 1, 1)$; LeakyReLU; $\mathrm{Conv}(32, 32, 3, 1, 1)$; LeakyReLU; $\mathrm{SE}(256, 32, 4)$
Layer 12	$\mathrm{Conv}(32, S \times T \times 2, 3, 1, 1)$
Output	Inter-motion map $\boldsymbol{n}$

Table S4. The structure of inter-motion analysis. $\mathrm{Up}(\cdot)$ stands for bilinear interpolation.

Input	A sequence of blurred frames ($\mathrm{indim} = T \times 3$)
Layer 1	$\mathrm{Conv}(\mathrm{indim}, 64, 7, 1, 3)$; LeakyReLU; $\mathrm{ResBlock}(64, 64, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 3$
Layer 2	$\mathrm{Conv}(64, 128, 3, 2, 1)$; LeakyReLU; $\mathrm{ResBlock}(128, 128, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 6$
Layer 3	$\mathrm{Conv}(128, 256, 3, 2, 1)$; LeakyReLU; $\mathrm{ResBlock}(256, 256, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 6$
Layer 4	$\mathrm{Conv}(256, 256, 3, 2, 1)$; LeakyReLU; $\mathrm{ResBlock}(256, 256, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 6$
Layer 5	$\mathrm{TransConv}(256, 256, 3, 2, 1)$; LeakyReLU; $\mathrm{ResBlock}(256, 256, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 6$
Layer 6	$\mathrm{E\_Conv}(256, 256, 4)$
Layer 7	$\mathrm{Motion\_Refine}(256, 3, S \times T)$; $\mathrm{Up}(2)$
	Addition of outputs from *Layer 6* and *Layer 3*
Layer 8	$\mathrm{TransConv}(256, 128, 3, 2, 1)$; LeakyReLU; $\mathrm{ResBlock}(128, 128, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 6$
Layer 9	$\mathrm{E\_Conv}(256, 128, 4)$
Layer 10	$\mathrm{Motion\_Refine}(128, 3, S \times T)$
	Addition of outputs from *Layer 10* and *Layer 7*
Layer 11	$\mathrm{Up}(2)$
	Addition of outputs from *Layer 9* and *Layer 2*
Layer 12	$\mathrm{TransConv}(128, 64, 3, 2, 1)$; LeakyReLU; $\mathrm{ResBlock}(64, 64, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 6$
Layer 13	$\mathrm{E\_Conv}(256, 64, 4)$
Layer 14	$\mathrm{Motion\_Refine}(64, 3, S \times T)$
Layer 15	Addition of outputs from *Layer 14* and *Layer 11*
	Addition of outputs from *Layer 13* and *Layer 1*
Layer 16	$\mathrm{ResBlock}(64, 64, 3, 1, 1, \mathrm{BN} = \mathrm{False}) \times 3$; $\mathrm{Conv}(64, S \times T \times 3, 3, 1, 1)$; $\mathrm{Split}(3)$
Layer 17	Addition of outputs from *Layer 16* and *Layer 15*
Output	Final sharp video sequence $\hat{\boldsymbol{x}}$

Table S5. The architecture of video reconstruction network. Adapting to $\boldsymbol{u}$ via exposure-aware convolution is in the form $\mathrm{E\_Conv}$ (vec channel, input channel, reduction). Motion refinement is in the form of $\mathrm{Motion\_Refine}$ (input channel, kernel size, output sequence). $\mathrm{Split}(\cdot)$ stands for the channel splitting operation.

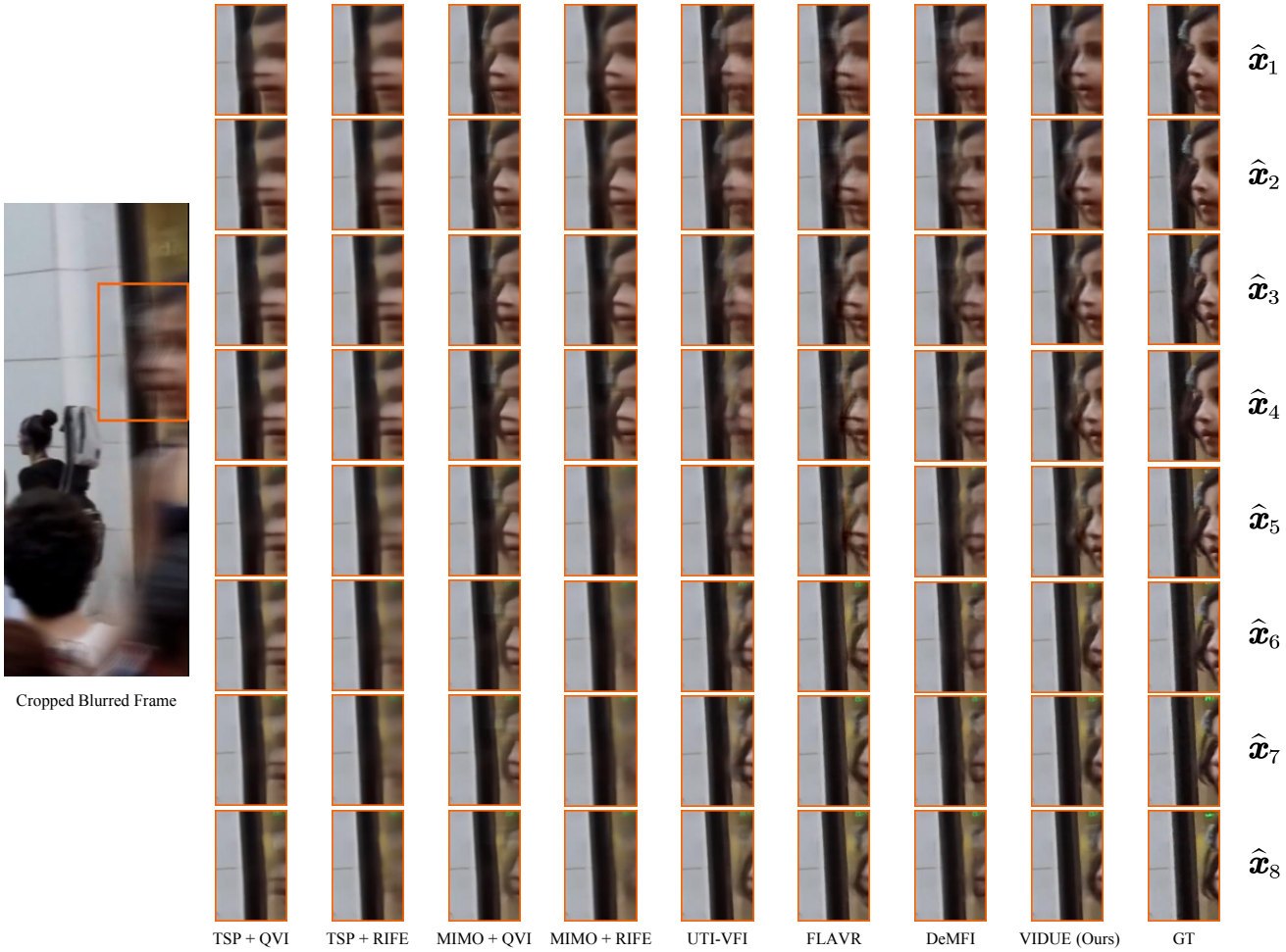

Figure S2. More visual results on GoPro [1] ($\times 8$).

Figure S3. More visual results on Adobe [2] ($\times 8$). Blue and red patches are from deblurred and interpolated frames, respectively.

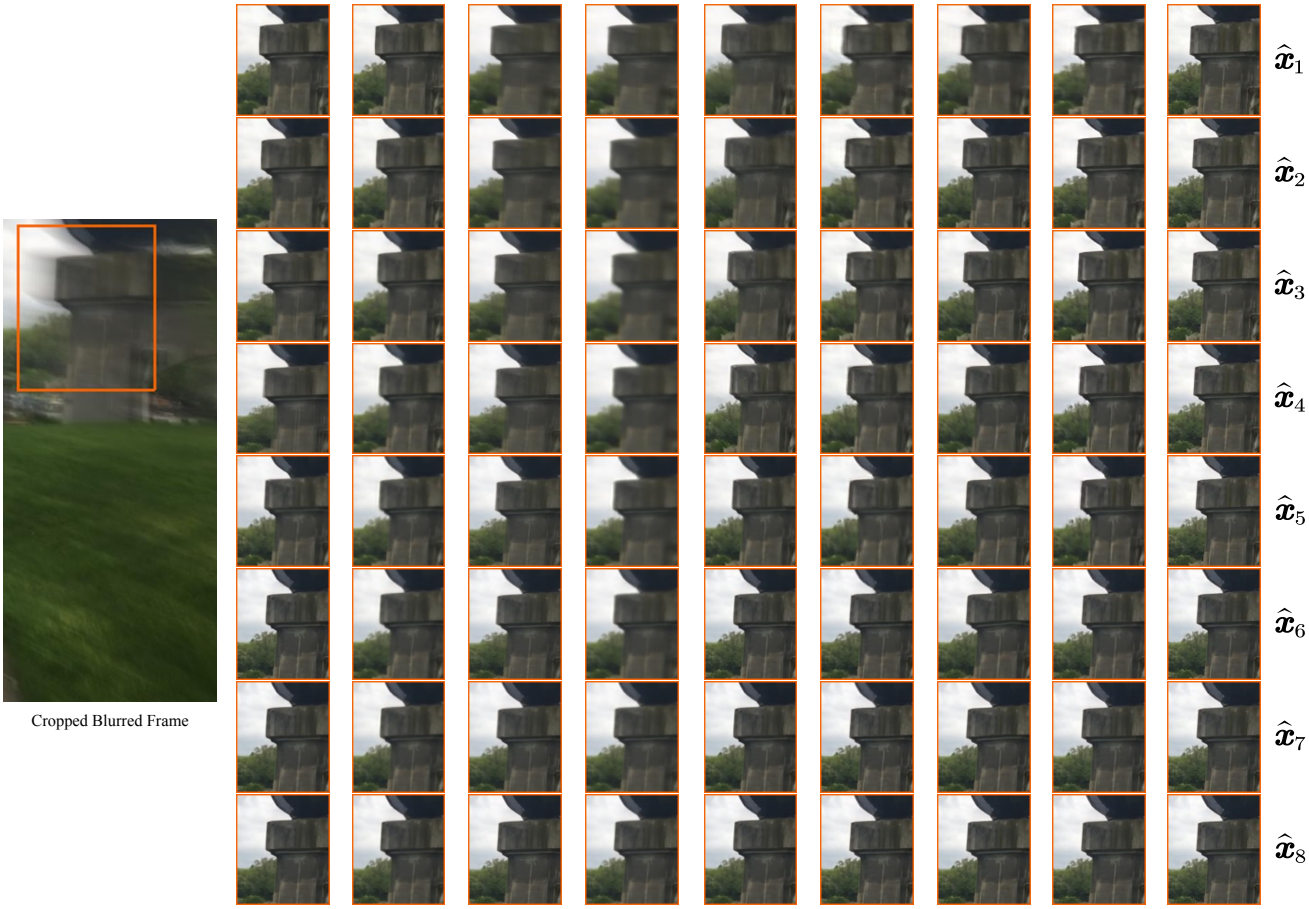

Figure S4. More visual results on Adobe [2] ($\times 8$).

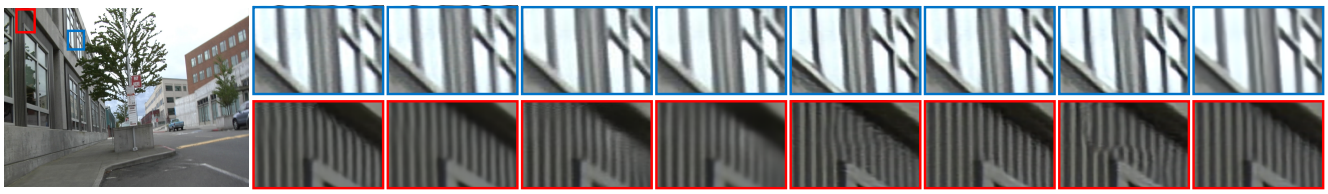

Figure S5. More visual results on RealBlur [2] ($\times 8$). Blue and red patches are from deblurred and interpolated frames, respectively.

Figure S6. More visual results on RealBlur [2] ($\times 8$).

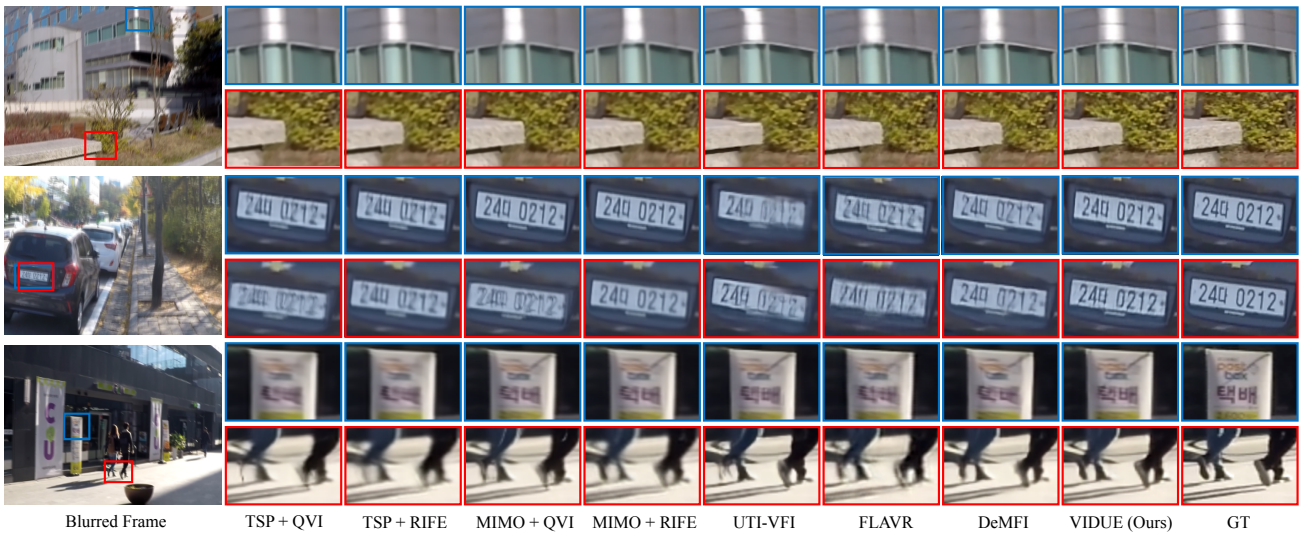

Figure S7. More visual results on GoPro [1] ($\times 16$). Blue and red patches are from deblurred and interpolated frames, respectively.

Cropped Blurred Frame

Figure S8. More visual results on GoPro [1] ($\times 16$).