DeMFI: Deep Joint Deblurring and Multi-Frame Interpolation with Flow-Guided Attentive Correlation and Recursive Boosting – Supplementary Material –

Jihyong Oh[0000−0002−1627−0529] and Munchurl Kim[0000−0003−0146−5419]

Korea Advanced Institute of Science and Technology, Daejeon, South Korea
{jhoh94, mkimee}@kaist.ac.kr

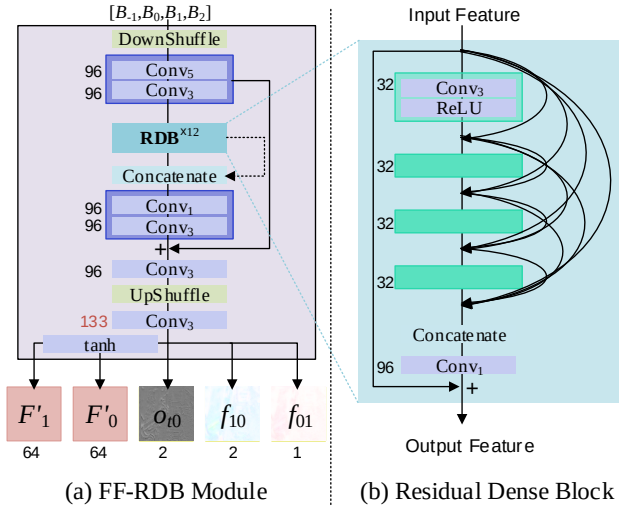

Fig. 1: Architecture of Feature Flow Residual Dense Backbone (FF-RDB) Module based on Residual Dense Block [16]. It is modified from [7,8] and DownShuffle layer distributes the motion information into channel axis [7,8].

1 Details of Architecture for DeMFI-Net

1.1 DeMFI-Net$_{bs}$

Feature Flow Residual Dense Backbone (FF-RDB) Module The feature flow residual dense backbone (FF-RDB) module first takes four consecutive blurry input frames (B_{-1}, B_0, B_1, B_2). It is similar to a backbone network of [8,7] and the number of output channels is modified to 133 ($= 64 \times 2 + 2 \times 2 + 1$). As shown in Fig. 1 (a), it consists of one DownShuffle layer and one UpShuffle layer [9], six convolutional layers, and twelve residual dense blocks [16] that are each composed of four Conv$_3$'s, one Conv$_1$, and four ReLU functions as in

Fig. 1 (b). All the hierarchical features obtained by the residual dense blocks are concatenated for successive network modules. The 133 output channels are composed of 64×2 for two feature maps (F_0', F_1') followed by tanh activation functions, 2×2 two bidirectional feature-domain flows (f_{01}, f_{10}) and 1 for an occlusion map logit (o_{t0}).

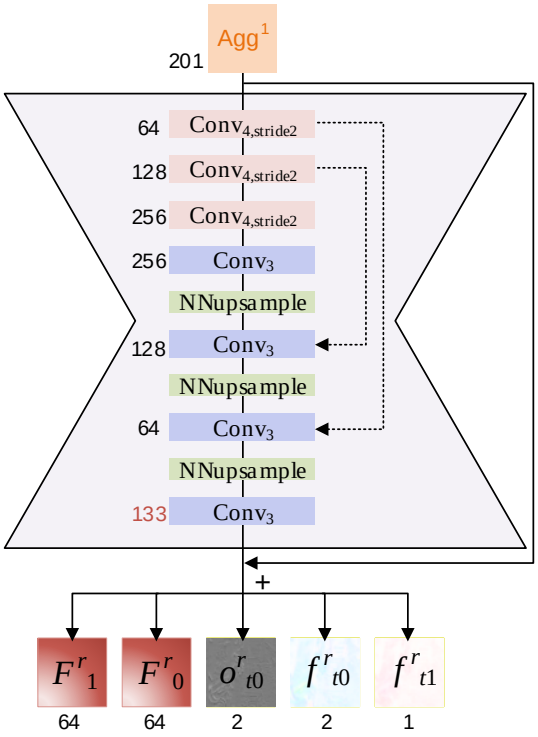

Fig. 2: Architecture of U-Net-based Refine Module (RM). NNupsample denotes nearest neighborhood upsampling.

U-Net-based Refine Module (RM) The U-Net-based [6] Refine Module (RM) in Fig. 2 takes $\mathbf{Agg}^1$ as an input to refine F_0^b, F_1^b, f_{t0}, f_{t1} and o_{t0} in a residual learning manner as $[F_0^r, F_1^r, f_{t0}^r, f_{t1}^r, o_{t0}^r] = \mathrm{RM}(\mathbf{Agg}^1) + [F_0^b, F_1^b, f_{t0}, f_{t1}, o_{t0}]$ where $\mathbf{Agg}^1$ is the aggregation of $[F_0^b, F_t, F_1^b, f_{t0}, f_{t1}, o_{t0}, f_{01}, f_{10}]$ in the concatenated form.

1.2 DeMFI-Net$_{rb}$

Booster Module Booster Module iteratively updates $\mathbf{f_P}$ to perform PWB for S_0^r, S_1^r obtained from DeMFI-Net$_{bs}$. The Booster Module is composed of Mixer and GRU-based Booster (GB), and it first takes a recurrent hidden state

(F_{i-1}^{rec}) and $\mathbf{f_P}^{i-1}$ at i-th recursive boosting as well as an aggregation of several components in the form of $\mathbf{Agg}^2 = [S_0^r, S_t^r, S_1^r, B_{-1}, B_0, B_1, B_2, f_{01}, f_{10}, \mathbf{f_F}]$ as an input to yield two outputs of F_i^{rec} and $\mathbf{\Delta}_{i-1}$ that is added on $\mathbf{f_P}^{i-1}$. Note that $\mathbf{f_P^0} = \mathbf{f_F}$ and $\mathbf{Agg}^2$ is not related to i-th recursive boosting. The updating process is given as follows:

$$M_{i-1} = \text{Mixer}([\mathbf{Agg}^2, \mathbf{f_P}^{i-1}]) \tag{1}$$

$$[F_i^{rec}, \mathbf{\Delta}_{i-1}] = \text{GB}([F_{i-1}^{rec}, M_{i-1}]) \tag{2}$$

$$\mathbf{f_P}^{i} = \mathbf{f_P}^{i-1} + \mathbf{\Delta}_{i-1}, \tag{3}$$

where the initial feature F_0^{rec} is obtained as a 64-channel feature via channel reduction for $\text{Conv}_1([F_0^r, F_t^r, F_1^r])$ of 192 channels. More details about both Mixer and the updating process of GB are described in the following subsections.

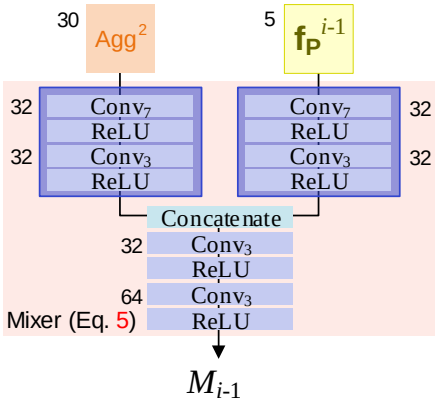

Fig. 3: Architecture of Mixer in Booster Module. It is designed to blend two information of $\mathbf{Agg}^2$ and $\mathbf{f_P}^{i-1}$. Eq. 5 means an equation in the main paper.

Mixer The first component in Booster Module is called Mixer. As shown in Fig. 3, Mixer first passes $\mathbf{Agg}^2$ and $\mathbf{f_P}^{i-1}$ through each independent set of convolution layers as $\text{Conv}_7 - \text{ReLU} - \text{Conv}_3 - \text{ReLU}$, respectively, then yields M_{i-1} via $\text{Conv}_3 - \text{ReLU} - \text{Conv}_3 - \text{ReLU}$ by taking concatenated outputs of the sets. M_{i-1} is consecutively used in GRU-based Booster (GB) as described in the following subsection.

GRU-based Booster (GB) GRU-based Booster (GB) takes both M_{i-1} and F_{i-1}^{rec} as an input to finally produce an updated F_i^{rec} which is consecutively used to make $\mathbf{\Delta}_{i-1}$ that is added on $\mathbf{f_P}^{i-1}$. GB adopts gated activation unit based on the GRU cell [1] by replacing fully connected layers with two separable convolutions of 1×5 ($\text{Conv}_{1\times5}$) and 5×1 ($\text{Conv}_{5\times1}$) as in [12] to efficiently

increase a receptive field. The detailed process in GB is operated as follows:

$$z_i^{1\times5} = \sigma(\text{Conv}_{1\times5}([F_{i-1}^{rec}, M_{i-1}])) \tag{4}$$

$$r_i^{1\times5} = \sigma(\text{Conv}_{1\times5}([F_{i-1}^{rec}, M_{i-1}])) \tag{5}$$

$$\hat{F}_i^{rec,1\times5} = \tanh(\text{Conv}_{1\times5}[r_i^{1\times5} \odot F_{i-1}^{rec}, M_{i-1}])) \tag{6}$$

$$F_i^{rec,1\times5} = (1 - z_i^{1\times5}) \odot F_{i-1}^{rec} + z_i^{1\times5} \odot \hat{F}_i^{rec,1\times5} \tag{7}$$

$$z_i^{5\times1} = \sigma(\text{Conv}_{5\times1}([F_i^{rec,1\times5}, M_{i-1}])) \tag{8}$$

$$r_i^{5\times1} = \sigma(\text{Conv}_{5\times1}([F_i^{rec,1\times5}, M_{i-1}])) \tag{9}$$

$$\hat{F}_i^{rec,5\times1} = \tanh(\text{Conv}_{5\times1}([r_i^{5\times1} \odot F_i^{rec,1\times5}, M_{i-1}])) \tag{10}$$

$$F_i^{rec} = (1 - z_i^{5\times1}) \odot F_i^{rec,1\times5} + z_i^{5\times1} \odot \hat{F}_i^{rec,5\times1} \tag{11}$$

$$\mathbf{\Delta}_{i-1} = (\text{Conv}_3 \circ \text{RL} \circ \text{Conv}_3)(F_i^{rec}). \tag{12}$$

Please note that Eqs. (11) and (12) produce the final outputs (F_i^{rec}, $\mathbf{\Delta}_{i-1}$) of the Booster Module as shown in Fig. 4 in the main paper, indicated by blue arrows.

2 Additional Qualitative Comparison Results

Figs. 4, 5, 6, 7, 8 show the abundant visual comparisons of deblurring and MFI (×8) performances for all the three test datasets. To better show them, we generally show the cropped patches for each scene. Since the number of blurry input frames for each method is different, two blurry center-input frames (B_0, B_1) are averagely shown in the figures. As shown, the severe blurriness can easily be shown between two center-input frames (B_0, B_1), which is very challenging for VFI.

Our DeMFI-Nets, especially DeMFI-Net$_{rb}$, better synthesize textures or patterns (1st/2nd scenes of Fig. 4, Fig. 5, 1st scene of Fig. 8), precisely generate thin poles (3rd scene of Fig. 4) or fast moving objects (2nd/3rd scenes of Fig. 7) and effectively capture letters (Fig. 5, Fig. 6, 1st scene of Fig. 7, 2nd/3rd/4th scenes of Fig. 8), which tend to be failed by all the previous methods.

Especially, CFI methods such as TNTT and PRF are more hard to interpolate sharp frames at the time index 2/8 or 6/8 than 4/8 (center time instance) within each scene because they can only produce intermediate frames of time at a power of 2 in a recursive manner. As a result, the prediction errors are accumulatively propagated to the later interpolated frames. On the other hand, our DeMFI-Net framework adopts self-induced flow-based warping methodology trained in an end-to-end manner, which finally leads to generate *temporally consistent* sharp intermediate frames from blurry input frames. Also the results of deblurring and MFI (×8) of all the SOTA methods are publicly available at https://github.com/JihyongOh/DeMFI for easier comparison. Please note that it is laborious but worth to get results for the SOTA methods in terms of MFI (×8).

3 Limitations: Failure Cases

Extreme conditions such as tiny objects, low-light condition and large motion would make the joint task very challenging. Fig. 9 shows the failure cases such as tiny objects (1st scene), low-light condition (2nd scene) and large motion (3rd scene), which would make the joint task very challenging. First, in the case of splashed tiny objects with blurriness, it is very hard to capture sophisticated motions from the afterimages of the objects so all the methods fail to delicately synthesize the frames as GT's. Second, in the case of low-light condition, it is hard to distinguish the boundaries of the objects (green arrows) and to detect tiny objects such as fast falling coffee beans (dotted green line), which deteriorates the overall performances of all the methods. Lastly, large and complex motion with blurriness due to camera shaking also makes all the methods hard to precisely synthesize final frames as well. In addition, as shown in Fig. 10, misled attention further affect poor final results. We hope these kinds of failure cases will motivate researchers for further challenging studies.

4 Visual Comparison with Demo Video

We provide a visual comparison video at https://youtu.be/J93tW1uwRy0 for TNTT [5], UTI-VFI* (retrained ver.) [15], PRF [8] (a larger-sized version of [7]) and DeMFI-Net$_{rb}$ (5,3) (ours), which all have adopted joint learning for deblurring and VFI. The demo video shows several multi-frame interpolated ($\times$8) results played as 30fps for a slow motion, synthesized from blurry input frames of 30fps. All the results of the methods are adequately resized to be simultaneously played at a single screen. Please take into account that YouTube240 test dataset contains extreme motion with blurriness.

TNTT generally synthesize blurry visual results and PRF tends to show temporal inconsistency for MFI ($\times$8). These two joint methods simply do CFI, not for arbitrary time t. Therefore, their methods must be recursively applied after each center frame is interpolated for MFI, which causes error propagation into later-interpolated frames. Although UTI-VFI* shows better visual results than above two CFI joint methods, it tends to produce some artifacts especially on large motion with blurriness and tiny objects such as splash of water. This tendency is attributed to the error accumulation from the dependency on f_P quality inevitably obtained by pretrained PWC-Net [11], where adoption of a pretrained net also brings a disadvantage in terms of both R_t and #P (+8.75M). On the other hand, our DeMFI-Net framework is based on the self-induced feature- and pixel-domain flows without any help of pretrained optical flow networks, to finally better interpolate the sharp frames.

5 Number of Inputs

On the other hand, we trained DeMFI-Net$_{rb}$(5,3) with only two input frames, called DeMFI-Net$_{2f}$, during model development. This model yielded an average PSNR/SSIM/tOF of 33.23/0.9376/0.498 for Adobe240, 32.20/0.9228/0.485

for YouTube240 and 30.21/0.8957/0.546 for GoPro240, which were much inferior than original DeMFI-Net$_{rb}$(5,3) trained with 4 input frames as shown in Table 2 of main paper. Simply taking more blurry input frames strongly helps the network to capture useful latent information such as a temporal tendency and a motion information due to accumulation of light [2,3,13], which finally increases the joint performance. It is also worth to note that DeMFI-Net$_{2f}$ still outperforms all the previous methods in Table 2 of main paper.

6 Bolstered Features into FWB

We newly trained DeMFI-Net$_{bs}$ under a setting of feeding bolstered features into FWB, which results in 33.83/0.9410/0.471 for Adobe240, 32.83/0.9263/0.467 for YouTube240 and 30.76/0.9009/0.537 for GoPro240. These results are better than those of the original DeMFI-Net$_{bs}$ so it can be concluded that the placing the bolstered features earlier is also important to improve joint performances.

7 Arbitrary Time $\times M$

Fig. 11 shows qualitative interpolated results for $M = 3$, which supports the statement for arbitrary interpolation of our DeMFI-Net. There is also a 30fps demo video for $M = 5, 13, 19$ at https://github.com/JihyongOh/DeMFI.

8 Occlusion Maps

Occlusion maps have continuous values due to sigmoid, so features/images at both time indices (0, 1) jointly contribute to those at t by adopting Eq. (2) (in main paper) which is widely used in VFI methods [4,10,14]. Training tends to diverge without a constraint of $o_{t1} = 1 - o_{t0}$ [4]. Fig. 12 shows several components in PWB corresponding to Fig. 8 in main paper, when person moves fast to the right. It should also be noted that both (1-t) and t are respectively multiplied on each term in PWB to reflect influences according to temporal distances. On the other hand, the subscript t of o_{t0} can be dropped for a first FWB because it does not related to t for the first FWB, but should be kept for second FWB and PWB.

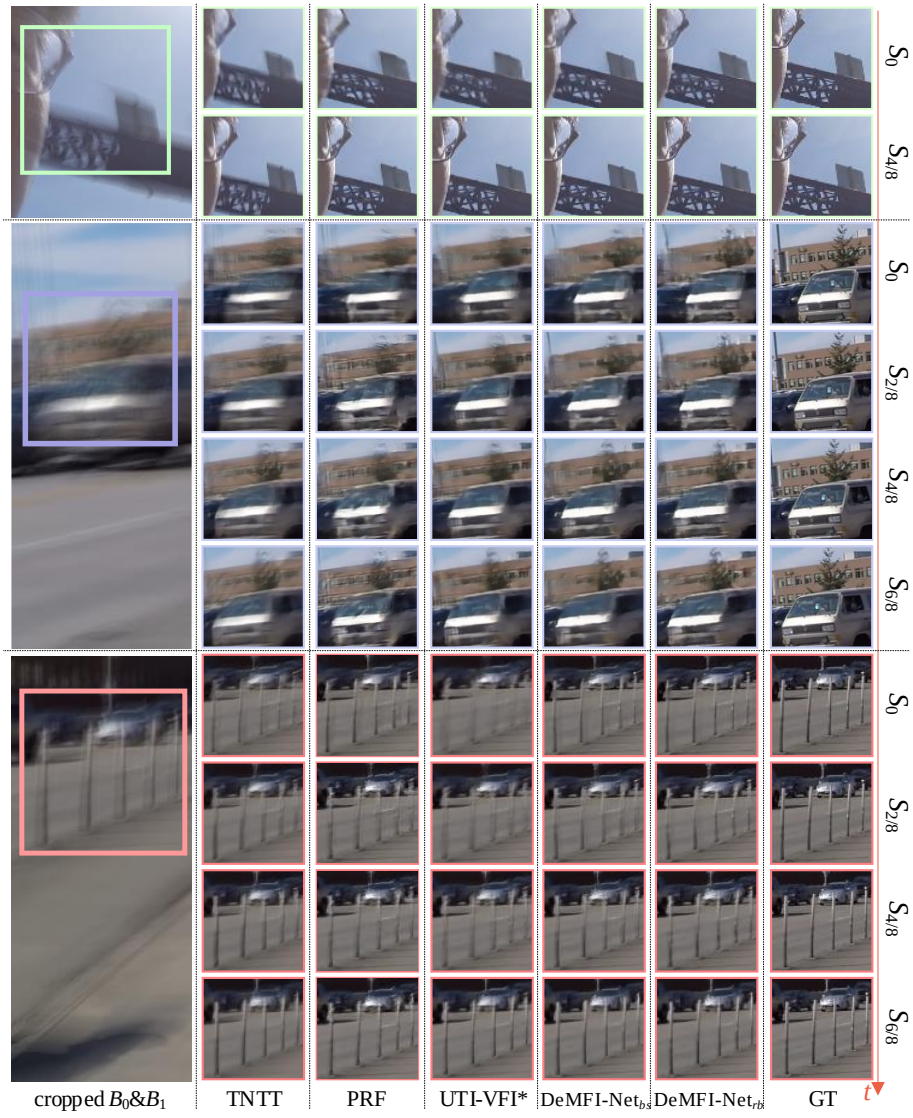

Fig. 4: Visual comparisons for MFI results on Adobe240. *Best viewed in zoom.*

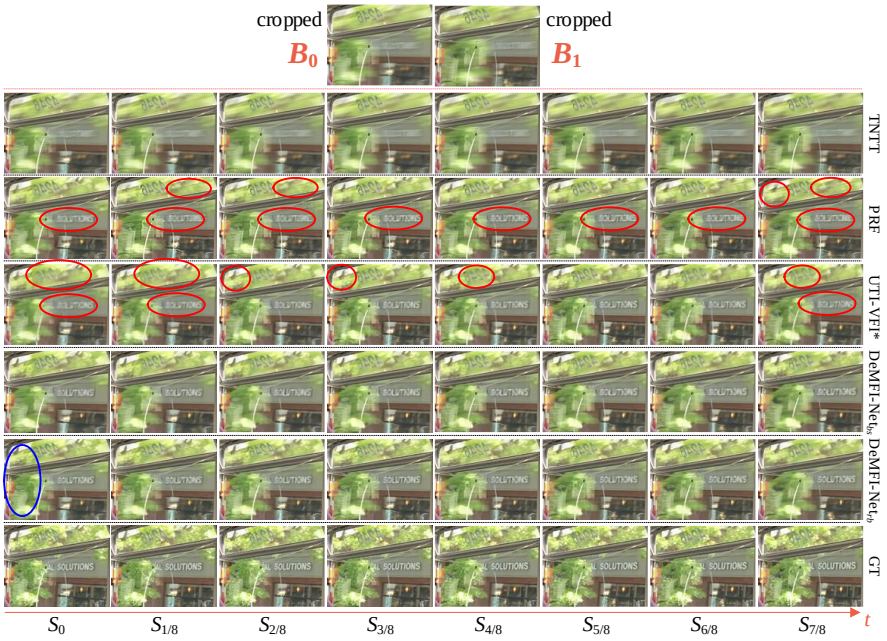

Fig. 5: Visual comparisons for MFI results on Adobe240. *Best viewed in zoom.*

Fig. 6: Visual comparisons for MFI results on GoPro240. *Best viewed in zoom.*

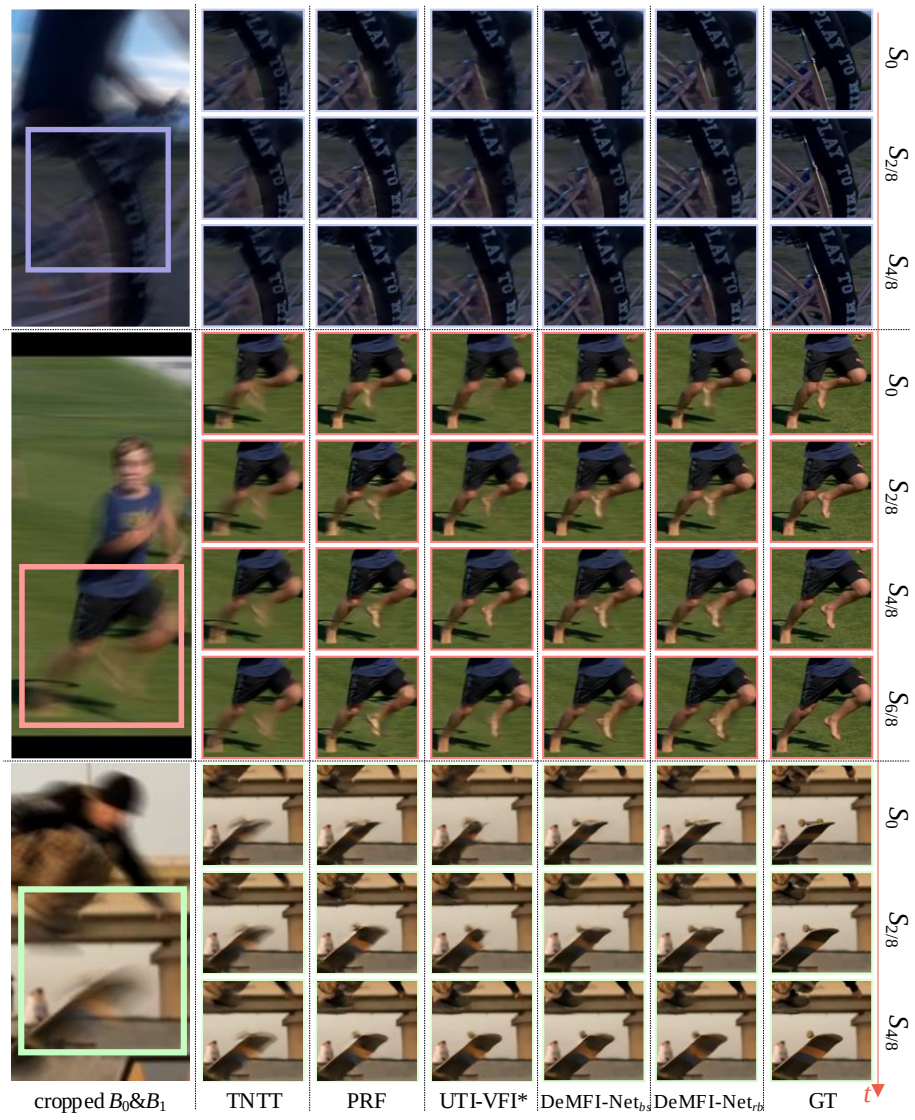

Fig. 7: Visual comparisons for MFI results on YouTube240. *Best viewed in zoom.*

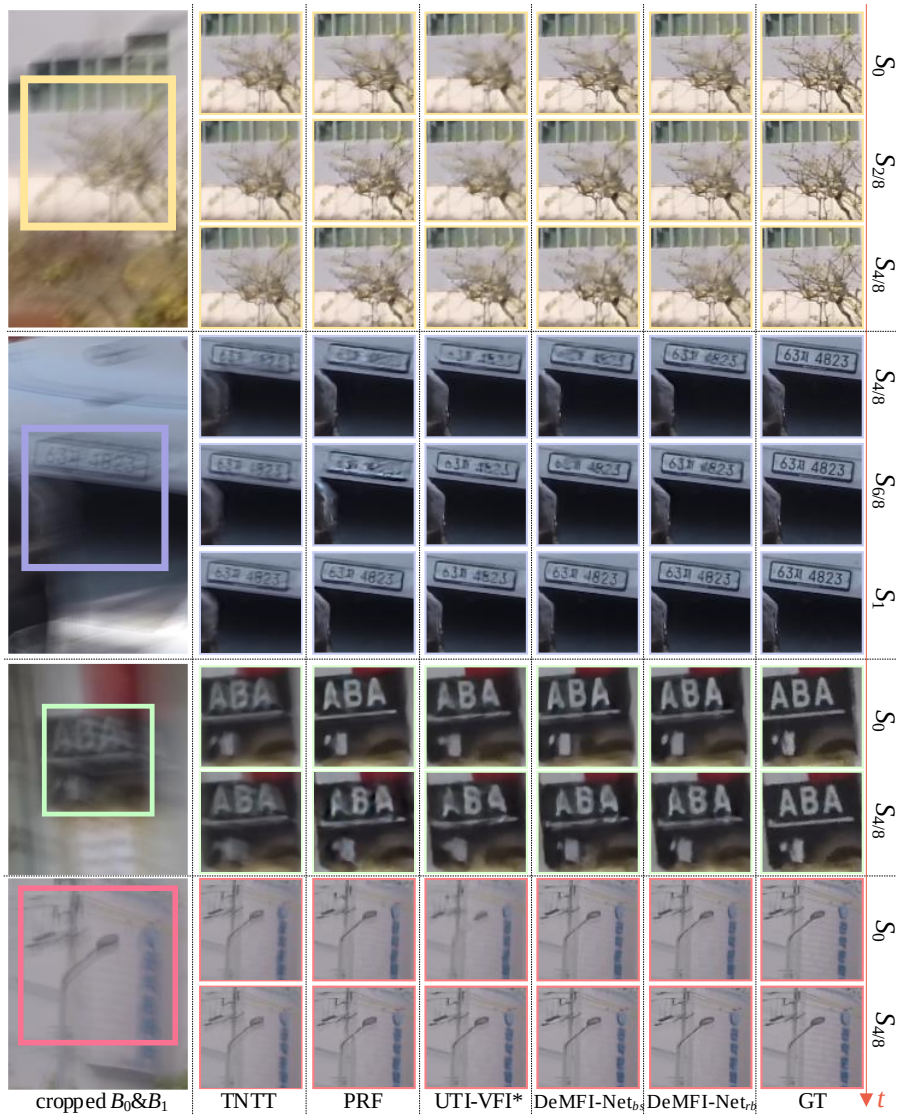

Fig. 8: Visual comparisons for MFI results on GoPro240. *Best viewed in zoom.*

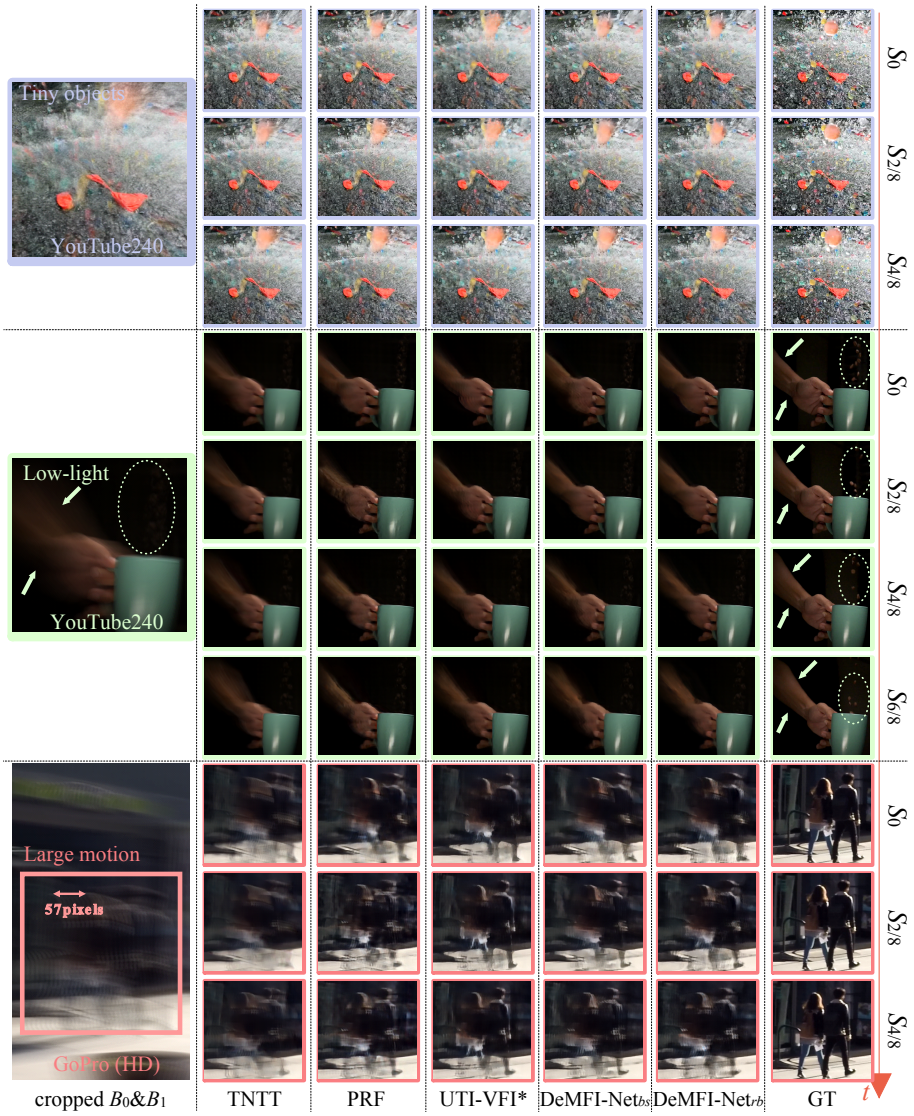

Fig. 9: Failure cases; tiny objects, low-light condition and large motion. *Best viewed in zoom.*

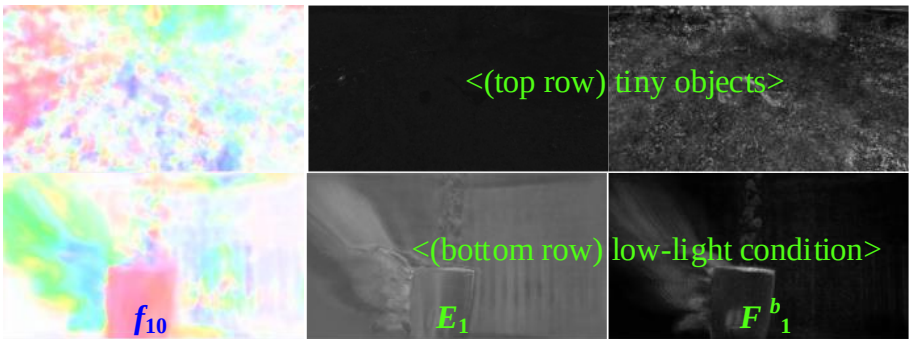

Fig. 10: Two failure cases in extreme conditions, corresponding to Fig. 9 for $S_{4/8}$. As shown, misled attention further affect poor final results.

Fig. 11: Qualitative interpolated results for $M = 3$. *Best viewed in zoom.*

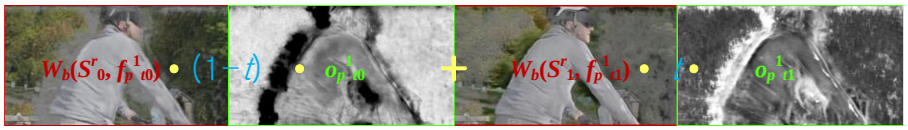

Fig. 12: Several components in PWB corresponding to Fig. 8 in main paper, when person moves fast to the right. It should be noted that both (1-t) and t are respectively multiplied on each term in PWB to reflect influences according to temporal distances. *Best viewed in zoom.*

References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP 3
2. Gupta, A., Joshi, N., Zitnick, C.L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: ECCV. pp. 171–184. Springer (2010) 6
3. Harmeling, S., Michael, H., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. NeurIPS **23**, 829–837 (2010) 6
4. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: CVPR. pp. 9000–9008 (2018) 6
5. Jin, M., Hu, Z., Favaro, P.: Learning to extract flawless slow motion from blurry videos. In: CVPR. pp. 8112–8121 (2019) 5
6. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 2
7. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Blurry video frame interpolation. In: CVPR. pp. 5114–5123 (2020) 1, 5
8. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Video frame interpolation and enhancement via pyramid recurrent framework. IEEE Transactions on Image Processing **30**, 277–292 (2020) 1, 5
9. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR. pp. 1874–1883 (2016) 1
10. Sim, H., Oh, J., Kim, M.: Xvfi: extreme video frame interpolation. In: ICCV (2021) 6
11. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR. pp. 8934–8943 (2018) 5
12. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV. pp. 402–419. Springer (2020) 3
13. Telleen, J., Sullivan, A., Yee, J., Wang, O., Gunawardane, P., Collins, I., Davis, J.: Synthetic shutter speed imaging. In: Computer Graphics Forum. vol. 26, pp. 591–598. Wiley Online Library (2007) 6
14. Xu, X., Siyao, L., Sun, W., Yin, Q., Yang, M.H.: Quadratic video interpolation. In: NeurIPS. pp. 1647–1656 (2019) 6
15. Zhang, Y., Wang, C., Tao, D.: Video frame interpolation without temporal priors. NeurIPS **33** (2020) 5
16. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. pp. 2472–2481 (2018) 1