DSF-Net: Dual-Stream Fused Network for Video Frame Interpolation

Fuhua Zhang and Chuang Yang

***Abstract*—Video frame interpolation aims to improve the video quality by increasing the frame rate. Existing methods adopt the cascaded architecture. They first estimate intermediate flow maps and then refine the synthesized intermediate frames with contextual features separately. However, the separated flow estimation and refined module ignore the mutual facilitation of them in frame interpolation. Following this issue, we propose a Dual-Stream Fused Network (DSF-Net) to joint flow estimation and refinement module for frame interpolation. Specifically, it first extracts the contextual features from input frames by a contextual feature extractor module, and then jointly refines the intermediate flow maps with the extracted features through a coarse-to-fine frame synthesis module. DSF-Net allows the intermediate flow and the contextual features to benefit each other while generating sharper moving objects and capturing better textual details. Experimental results demonstrate that DSF-Net performs consistently better than existing SOTA methods.**

***Index Terms*—Video frame interpolation, Coarse-to-fine, Dual-stream fused.**

I. Introduction

VIDEO Frame Interpolation (VFI) [1] aims to up-convert low frame rate video to high frame rate. VFI enhances the detail of changes from the previous frame to the next by interpolating motion information. Therefore, VFI can be used for a range of video applications, including frame rate up-conversion [2], [3], video compression [4], [5], motion blur generation [6] and cartoon creation [7].

Existing mainstream VFI methods can be approximately classified into two folds, flow-based [8], [9], [10], [11], [12] and kernel-based [13], [14], [15], [16], [17] methods. Motion estimation and frame synthesis are two main steps in VFI methods. Flow-based methods employ optical flow to estimate the object motion first, and then synthesize the intermediate frames according to the computed intermediate flow maps. Without using the optical flow, kernel-based methods combine motion estimation and frame synthesis into an end-to-end architecture to directly synthesize the intermediate frames. In addition, a novel model [18], [19] has been proposed. It combines video frame interpolation and super resolution [20] to enhance video spatial-temporal quality. Specifically, Xiao *et al.* [21] proposed using temporal information to get more efficient spatial–temporal information for super-resolution.

Due to optical flow [22], [23] providing an unambiguous pixel-level motion correspondence between input frames, flow-based methods have become the most popular and achieved impressive results. The successful methods usually include two types, which one is shown in Fig. 1(a) [24], [25], [26], [27], [28]. It includes three steps:1) Estimating the intermediate flow maps. 2) Extracting contextual features of input frames. 3) Synthesizing intermediate frame with intermediate flow maps and contextual features. The other type is shown in Fig. 1(b) [29], [30], [31], [32], [33], [34]. It includes two steps:1) generating the coarse intermediate frame according to the estimated bilateral flow maps. 2) refining the intermediate frame by employing the contextual features. However, there are existing two defects in those methods. Firstly, the cascaded architecture ignores the mutual facilitation between intermediate flow and contextual features in frame interpolation. Secondly, the architectures are complex, which prevents them from real-time applications.

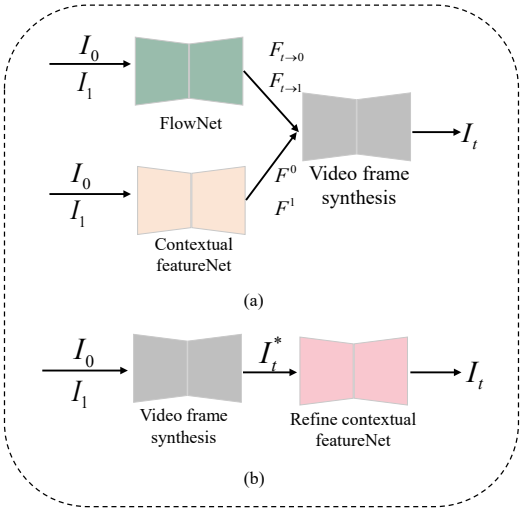

Fig. 1. Two Types of flow-based VFI methods. In (a), the intermediate flow maps $F_{t\to0}$, $F_{t\to1}$ and contextual features F^0, F^1 are acquired from the input frames I_0, I_1 respectively. $F_{t\to0}$, $F_{t\to1}$ and F^0, F^1 are employed to synthesis the intermediated frame I_t. In (b), the coarse intermediate frame I_t^* is generated through a video frame synthsis network. Then, the I_t^* is refined by contextual features to get the intermediate frame I_t.

To alleviate those issues, a Dual-Stream Fused Network (DSF-Net) is proposed for VFI task. It fuses the separated stream flow estimation and contextual feature extraction module into a single model. As showed in Fig. 2, it first extracts the multi-scale contextual features from input frames by an extractor, and then jointly refines the bilateral intermediate flow maps with the extracted features and the residual intermediate flow through a coarse-to-fine frame synthesis module. It can benefit the intermediate flow and contextual features with each other,

Fuhua Zhang and Chuang Yang are with the School of Computer Science, and with the School of Artificial Intelligence, OPtics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China. E-mail: fuhuazhang@mail.nwpu.edu.cn, omtcyang@gmail.com.

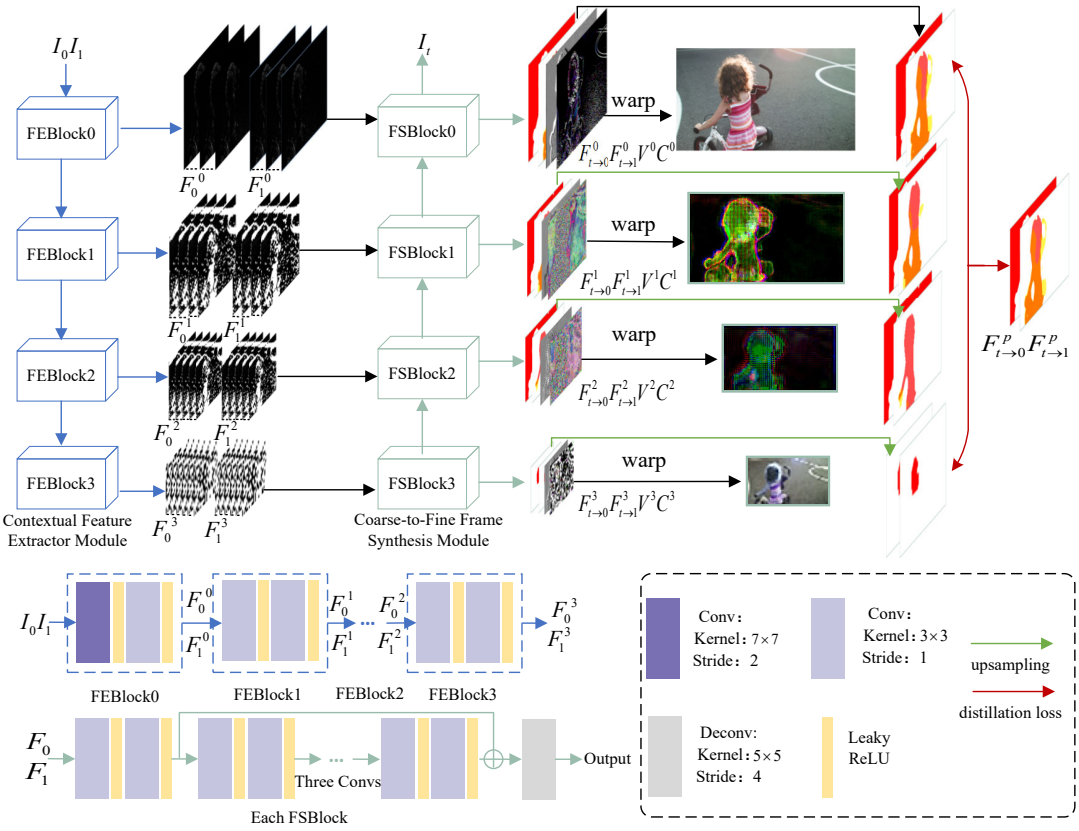

Fig. 2. Overview of the proposed DSF-Net. It is composed of two parts, *i.e.*, the contextual feature extractor and coarse-to-fine frame synthesis module.

so that enable our model generate high quality intermediate frame with sharping moving objects and better textual details. Specifically, there are four blocks in the contextual feature extractor and coarse-to-fine frame synthesis module. To handle the large motion encountered in frame interpolation, we first compute the rough intermediate flow maps on the low scale features, then iteratively refine the flow maps with gradually increasing scales. Furthermore, an intermediate flow distillation loss is designed in DSF-Net. It is used to guide multi-scale intermediate flow estimation, which can boost the accuracy of the flow maps. The differences between DSF-Net and Prost-Net [34] are as follows: (1) Prost-Net has two branches, one is used to extract temporal information and the other is used to get spatial information. Then, the two information are fused to generate intermediate frames. (2) DSF-Net only has one branch, which uses contextual information to generate intermediate flow maps and then generate intermediate frames.

In this paper, the contributions of the proposed method can be summarized:

(1) DSF-Net is proposed to joint intermediate flow estimation and contextual features refinement for video frame interpolation, which can generate sharper moving objects and capture better textual details.

(2) The intermediate flow distillation loss is designed to supply deep supervision and effectively guide intermediate flow estimation, which can boost the accuracy of intermediate flow maps.

(3) Experimental results show that the proposed method DSF-Net performs consistently better than existing state-of-the-art methods.

II. The Proposed Method

A. Overview

Given a pair of consecutive frames I_0 and I_1 from a low frame-rate video, the proposed Dual-Stream Fused Network (DSF-Net) aims to synthesize an intermediate frame I_t. It fuses synthesis and refined modules into a single network. As shown in Fig. 2, it is comprised of two parts, *i.e.*, the contextual feature extractor and coarse-to-fine frame synthesis module. The following is the detailed presentation of each part.

B. Contextual Feature Extractor Module

The contextual features extractor contains four feature extraction blocks (FEBlock), which respectively extract features of different scales. The inputs into the contextual feature extractor are the two input frames I_0 and I_1, and the outputs are the extracted contextual features with different scales as:

$$F_0^0 = \text{FEBlock0}(I_0), F_0^i = \text{FEBlock}i(F_0^i), (i = 1, 2, 3), \quad (1)$$

$$F_1^0 = \text{FEBlock0}(I_1), F_1^i = \text{FEBlock}i(F_1^i), (i = 1, 2, 3), \quad (2)$$

where F_0^i and F_1^i (i = 0,1,2,3) represent the four different scales contextual features *i.e.* one-half, one-quarter, one-eighth and one-sixteenth.

C. Coarse-to-Fine Frame Synthesis Module

After getting the extracted contextual features from the extractor module, we then estimate intermediate frame maps and refine them with the extracted features in a coarse-to-fine frame synthesis module. It includes four frame synthesis blocks (FSBlock). Specifically, the inputs of each FSBlock are the extracted contextual features F_0^i and F_1^i, and the outputs are two branches, the first are intermediate flow maps $F_{t\to0}^i$ $F_{t\to1}^i$. The second are the visibility map V^i and residual frame C^i, which are used to adjust and compensate for the occlusion information in frame interpolation.

The coarse-to-fine frame synthesis module first computes intermediate flow maps on the smaller contextual features, which is believed to capture large motions easier. We input F_0^3 and F_1^3 into FSBlock3, then get the intermediate flow maps, visibility map, and residual frame as:

$$F_{t\to0}^3, F_{t\to1}^3, V^3, C^3 = \mathrm{FSBlock3}(F_0^3, F_1^3). \tag{3}$$

The intermediate flow maps $F_{t\to0}^3$, $F_{t\to1}^3$, visibility maps V^3 and residual frame C^3 can be used to generate intermediate frame I_t^3 as:

$$I_t^3 = I_0^3 \odot V^3 + I_1^3 \odot (1 - V^3) + C^3, \tag{4}$$

$$I_0^3 = \mathrm{warp}(I_0^{\downarrow 8}, F_{t\to0}^3), \tag{5}$$

$$I_1^3 = \mathrm{warp}(I_1^{\downarrow 8}, F_{t\to1}^3). \tag{6}$$

where $\mathrm{warp}(\cdot,\cdot)$ represents the bilinear interpolation, $\odot$ is an element-wise multiplier, $\downarrow 8$ means down-sampling 8 times.

The obtained I_t^3, $F_{t\to0}^3$, $F_{t\to1}^3$, F_0^2, and F_1^2 are used to refine the intermediate flow maps. They are input to FSBlock2 and get the refined the intermediate flow maps $F_{t\to0}^2$, $F_{t\to1}^2$, visibility map V^2 and residual frame C^2 as:

$$F_{t\to0}^2, F_{t\to1}^2, V^2, C^2 = \mathrm{FSBlock2}(F_0^2, F_1^2, I_t^3, F_{t\to0}^3, F_{t\to1}^3) + \mathrm{Up}^2(\mathrm{F}_{t\to0}^3, \mathrm{F}_{t\to1}^3, \mathrm{V}^3, \mathrm{C}^3), \tag{7}$$

where Up^2 is upsampling, which is used to upsample the obtained intermediate flow, visibility map, and residual frame for residual concatenation to optimize those maps. Then, the generate intermediate frame I_t^2 can be obtained as:

$$I_t^2 = I_0^2 \odot V^2 + I_1^2 \odot (1 - V^2) + C^2, \tag{8}$$

$$I_0^2 = \mathrm{warp}(I_0^{\downarrow 4}, F_{t\to0}^2), \tag{9}$$

$$I_1^2 = \mathrm{warp}(I_1^{\downarrow 4}, F_{t\to1}^2). \tag{10}$$

As with the FSBlock2, we iteratively refine the flow fields with gradually increasing resolution in the rest FSBlock . The final refined intermediate flow maps $F_{t\to0}^0$, $F_{t\to1}^0$, visibility map V^0 and residual frame C^0 are gotten from FSBlock0. Then the intermediate frame I_t^0 is obtained as:

$$I_t^0 = I_0 \odot V^0 + I_1^0 \odot (1 - V^0) + C^0, \tag{11}$$

$$I_0^0 = \mathrm{warp}(I_0, F_{t\to0}^0), \tag{12}$$

$$I_1^0 = \mathrm{warp}(I_1, F_{t\to1}^0). \tag{13}$$

Lastly, the intermediate frames obtained from each FSBlock are fused together with upsampling connection. The final frame intermediate frame I_t is obtained as:

$$I_t = I_t^0 + \mathrm{Up}^2(I_t^1) + \mathrm{Up}^4(I_t^2) + \mathrm{Up}^8(I_t^3). \tag{14}$$

D. Optimization

In this paper, the proposed method includes synthesis loss L_{syn} and intermediate flow distillation loss L_{dis}. They are formulated as follows:

$$L = L_{syn} + L_{dis}. \tag{15}$$

Following previous work [30], L_{syn} measures the similarity of the synthesized frame and the ground truth. L_{syn} loss is formulated as follows:

$$L_{syn} = \rho(I_{GT} - I_t) + L_{cen}(I_{GT}, I_t). \tag{16}$$

where $\rho(\mathrm{x}) = (x^2 + \epsilon^2)^\alpha$ with $\alpha = 0.5$, $\epsilon = 10^{-3}$ is the Charbonnier loss [35]. While L_{cen} is the census loss, which calculates the soft Hamming distance between census-transformed [36] image patches of size 7×7.

Considering that the input to the proposed model has no supervision in the intermediate flow, an intermediate flow distillation loss L_{dis} is employed to promote intermediate motion estimation. Specifically, I_{GT}, I_0 and I_{GT}, I_1 are input to the pre-trained LiteFlownet [37] estimation network to get the intermediate flow prediction $F_{t\to0}^p$ $F_{t\to1}^p$. Then the intermediate flow distillation loss L_{dis} is defined as :

$$L_{dis} = \parallel F_{t\to0}^p - (F_{t\to0}^{i\uparrow}) \parallel_1 + \parallel F_{t\to1}^p - (F_{t\to1}^{i\uparrow}) \parallel_1, \tag{17}$$

where $F_{t\to0}^{i\uparrow}$, $F_{t\to1}^{i\uparrow}$ mean $F_{t\to0}^{1,2,3}$, $F_{t\to1}^{1,2,3}$ up-smapling 2, 4, 8 times.

III. Experiments

A. Experiment Setup

In this section, the datasets, training details and Evaluation metrics are firstly introduced. Then, we compare the proposed method with representative methods on various benchmarks. Finally, ablation studies are conducted to analyze the contributions of the proposed method.

1) Datasets: In this paper, Vimeo-90k dataset [38] is adopted for training. It includes 91,701 triplets, where each triplet contains three consecutive frames with the resolution of 448 × 256. The training set consists of 51,313 frame triples, and the rest are used for testing. During training, the data sets are augmented by random flipping and reversing. UCF101 [39] and Middlebury datasets are used to test in DSF-Net. The UCF101 dataset includes 379 triplets with the resolution of 256×256. The Middlebury dataset [40] is widely used for optical flow and video frame interpolation task. It contains 12 challenging cases with the resolution around 640×480.

2) Training details: PyTorch [41] is employed to implement the proposed. During training on Vimeo-90k dataset, the proposed method is optimized by AdamW with batch size of 4 for 300 epochs. We gradually reduce the learning rate from 1×10^{-4} to 1×10^{-5}. It uses cosine annealing [42] during the whole training process. Four GeForce RTX 3090 are used to train the proposed method.

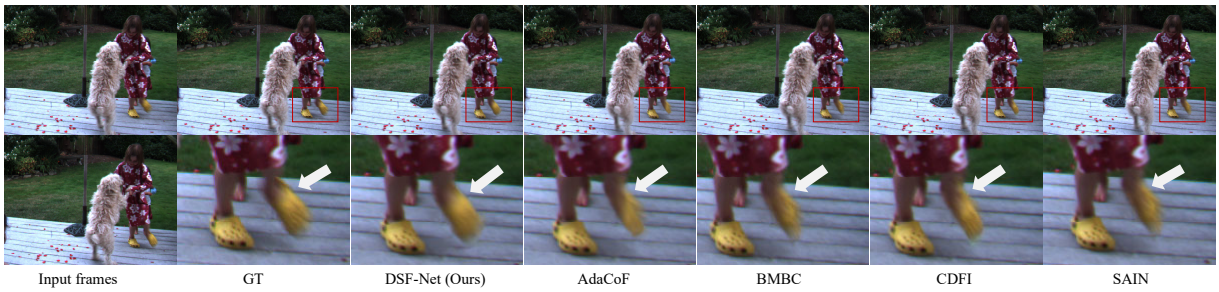

Fig. 3. Example results of different methods on the Middlebury dataset.

TABLE I
EFFECTIVENESS OF THE PROPOSED METHOD ON THE VIMEO-90K, UCF101, AND MIDDLEBURRY (MB) DATASETS.

Method	Vimeo-90k		UCF101		MB	Time	Params
	PSNR	SSIM	PSNR	SSIM	IE	(s)	(M)
AdaCoF [43]	34.47	0.973	34.91	0.968	2.24	0.271	21.84
BMBC [25]	35.01	0.976	35.15	**0.969**	2.04	1.603	11.0
CDFI [44]	35.17	0.964	35.21	0.950	1.98	0.064	5.0
ReMEI [26]	34.58	0.972	35.07	0.968	2.13	-	-
Sain [45]	35.74	0.979	35.20	**0.969**	2.08	-	-
EBME [46]	35.58	0.978	35.30	**0.969**	-	0.020	3.9
DSF-Net	**36.09**	**0.980**	**35.31**	**0.969**	**1.90**	**0.019**	17.87

3) Evaluation metrics: For quantitative evaluation, we use Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) and interpolation error (IE) metrics to evaluate the quality of synthesized intermediate frames. Higher PSNR and SSIM values mean the synthesized frame is closer to the ground truth. The metrics are positively correlated to the performance. The average interpolation error (IE) on the synthesized frames and the ground truth is also reported. The lower IE indicates the better performance.

B. Comparisons with State-of-the-art Methods

To verify the performance of the proposed method, we compare DSF-Net with state-of-the-art methods including AdaCoF [43], BMBC [25], CDFI [44], ReMEI-Net [26], Sain [45], and EBME [46]. Table I compares the PSNR, SSIM, IE, Times and Params evaluation metrics between the proposed DSF-Net and others. All methods are run on one GeForce RTX 3090 under 448×256 resolution for getting the inference speed. The proposed method DSF-Net only costs 0.019s achieving efficient frame interpolation. On the UCF101, Vimeo-90k and Middlebury datasets, the results prove the effectiveness of the proposed method. Furthermore, to better compare the results, the intermediate frames synthesized on the Middlebury by different methods are displayed in Fig. 3. It is clearly observed that DSF-Net can better preserve the object boundaries and eliminate the artifacts in boundaries. Overall, the results in Fig. 3 and Table I can jointly demonstrate the superiority and effectiveness of the proposed method.

C. Ablation Studies

In this section, the proposed method is analyzed by conducting ablation studies. Recall that an intermediate flow distillation loss is added to DSF-Net. To verify the contribution of the intermediate flow distillation loss to the DSF-Net, we compare the PSNR and SSIM values before and after its removal. Fig. 4 illustrates the visualization of intermediate flow maps with and with out the distillation loss. Fig. 4 shows the model with distillation loss generates the more detailed intermediate flow maps. As indicated in Table II, removing the intermediate flow distillation loss undermine the performance significantly. In addition, the performance of different blocks in DSF-Net is studied. Table II shows that DSF-Net with four blocks achieves the best results.

TABLE II
ABLATION STUDIES OF DSF-NET ON THE VIMEO-90K AND UCF101 DATASET.

Model	Vimeo-90k		UCF101		Time	Params
	PSNR	SSIM	PSNR	SSIM	(s)	(M)
w/o dis_loss	35.79	0.979	35.25	0.962	0.019	17.87
w/ 3blocks	35.33	0.975	35.21	0.960	0.009	16.47
w/ 5blocks	35.08	0.977	35.30	0.969	0.025	19.45
DSF-Net	36.09	0.980	35.31	0.969	0.019	17.87

Fig. 4. Ablation study on intermediate flow map $F^0_{t\to 0}$.

IV. CONCLUSION

In this paper, we propose a Dual-Stream Fused Network (DSF-Net) for VFI task. It fuses the flow estimation and refinement module into a single network for frame interpolation, which can allow the intermediate flow and the contextual features to benefit each other. Furthermore, an intermediate flow distillation loss is designed in DSF-Net. It is used to guide multi-scale intermediate flow estimation. Practically, the experiment results on frame interpolation task have demonstrated the excellence and effectiveness of the proposed method.

Although the intermediate frame quality and running time of the proposed DSF-Net have been greatly improved, we will be devoted to achieving a simpler framework for strong real-time performance and fewer parameters while maintaining high quality in the future.

REFERENCES

[1] Z. Yu, X. Chen, and S. Ren, "Video frame interpolation with learnable uncertainty and decomposition," *IEEE Signal Process. Lett.*, vol. 29, pp. 2642–2646, Dec. 2022.

[2] X. Yang, J. Liu, J. Sun, Y. Lee, and T. Q. Nguyen, "Depth-assisted frame rate up-conversion for stereoscopic video," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 423–427, Feb. 2014.

[3] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao, "High-order model and dynamic filtering for frame rate up-conversion," *IEEE Trans. on Image Process.*, vol. 27, no. 8, pp. 3813–3826, Apr. 2018.

[4] L. Dai and L. Zhang, "A joint spatiotemporal video compression based on stochastic adaptive fourier decomposition," *IEEE Signal Process. Lett.*, vol. 29, Jul. 2022.

[5] G. Lu, X. Zhang, L. Chen, and Z. Gao, "Novel integration of frame rate up conversion and hevc coding based on rate-distortion optimization," *IEEE Transactions on Image Process.*, vol. 27, no. 2, pp. 678–691, Oct. 2017.

[6] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6833–6841.

[7] L. Siyao, S. Zhao, W. Yu, W. Sun, D. Metaxas, C. C. Loy, and Z. Liu, "Deep animation video interpolation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6587–6595.

[8] Z. Shi, X. XU, X. Liu, J. Chen, and M.-H. Yang, "Video frame interpolation transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17 461–17 470.

[9] B. Zhao and X. Li, "Edge-aware network for flow-based video frame interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–8, Jun. 2022.

[10] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3703–3712.

[11] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9000–9008.

[12] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4463–4471.

[13] X. Ding, P. Huang, D. Zhang, and X. Zhao, "Video frame interpolation via local lightweight bidirectional encoding with channel attention cascade," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 1915–1919.

[14] S. Niklaus, L. Mai, and O. Wang, "Revisiting adaptive convolutions for video frame interpolation," in *Proc. IEEE WACV.*, 2021, pp. 1099–1109.

[15] Z. Shi, X. Liu, K. Shi, L. Dai, and J. Chen, "Video frame interpolation via generalized deformable convolution," *IEEE Trans Multimedia*, vol. 24, pp. 426–439, Jan. 2022.

[16] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 670–679.

[17] ——, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 261–270.

[18] M. Hu, K. Jiang, L. Liao, J. Xiao, J. Jiang, and Z. Wang, "Spatial-temporal space hand-in-hand: Spatial-temporal video super-resolution via cycle-projected mutual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3574–3583.

[19] Y. Xiao, Q. Yuan, J. He, Q. Zhang, J. Sun, X. Su, J. Wu, and L. Zhang, "Space-time super-resolution for satellite video: A joint framework based on multi-scale spatial-temporal transformer," *Int. J. Appl. Earth Obs. Geoinformation*, vol. 108, p. 102731, 2022.

[20] Y. Xiao, Q. Yuan, K. Jiang, J. He, Y. Wang, and L. Zhang, "From degrade to upgrade: Learning a self-supervised degradation guided adaptive network for blind remote sensing image super-resolution," *Inf Fusion*, vol. 96, pp. 297–311, 2023.

[21] Y. Xiao, X. Su, Q. Yuan, D. Liu, H. Shen, and L. Zhang, "Satellite video super-resolution via multiscale deformable convolution alignment and temporal grouping projection," *IEEE Trans. Geosci. Remote Sensing*, vol. 60, pp. 1–19, 2021.

[22] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "Flowformer: A transformer architecture for optical flow," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 668–685.

[23] X. Sui, S. Li, X. Geng, Y. Wu, X. Xu, Y. Liu, R. Goh, and H. Zhu, "Craft: Cross-attentional flow transformer for robust optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17 602–17 611.

[24] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 933–948, Sept. 2021.

[25] J. Park, K. Ko, C. Lee, and C. S. Kim, "Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 109–125.

[26] M. Hu, J. Xiao, L. Liao, Z. Wang, C.-W. Lin, M. Wang, and S. Satoh, "Capturing small, fast-moving objects: Frame interpolation via recurrent motion enhancement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3390–3406, Sept. 2022.

[27] D. M. Argaw, J. Kim, F. Rameau, and I. S. Kweon, "Motion-blurred video interpolation and extrapolation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 901–910.

[28] P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko, "Many-to-many splatting for efficient video frame interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3553–3562.

[29] H. Sim, J. Oh, and M. Kim, "Xvfi: Extreme video frame interpolation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14 489–14 498.

[30] J. Park, C. Lee, and C.-S. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14 539–14 548.

[31] D. M. Argaw and I. S. Kweon, "Long-term video frame interpolation via feature propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3533–3542.

[32] D. Danier, F. Zhang, and D. Bull, "St-mfnet: A spatio-temporal multi-flow network for frame interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3511–3521.

[33] K. Zhou, W. Li, X. Han, and J. Lu, "Exploring motion ambiguity and alignment for high-quality video frame interpolation," *arXiv preprint arXiv:2203.10291*, 2022.

[34] M. Hu, K. Jiang, L. Liao, Z. Nie, J. Xiao, and Z. Wang, "Progressive spatial-temporal collaborative network for video frame interpolation," in *ACM Multimedia*, 2022, pp. 2145–2153.

[35] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, 1994, pp. 168–172.

[36] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[37] T. Hui, X. Tang, and C. L. Change Loy, "A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8981–8989.

[38] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, 2019.

[39] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," pp. 107–123, 2019.

[40] T. Ha, S. Lee, and J. Kim, "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Trans on Consumer Electr.*, vol. 50, no. 2, pp. 752–759, May. 2004.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[42] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[43] H. Lee, T. Kim, T.-y. Chung, D. Pak, Y. Ban, and S. Lee, "Adacof: adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5316–5325.

[44] T. Ding, L. Liang, Z. Zhu, and I. Zharkov, "Cdfi: Compression-driven network design for frame interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8001–8011.

[45] Y. Lv, W. Yang, W. Zuo, Q. Liao, and R. Zhu, "Sain: Similarity-aware video frame interpolation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 1920–1924.

[46] X. Jin, L. Wu, G. Shen, Y. Chen, J. Chen, J. Koo, and C.-h. Hahm, "Enhanced bi-directional motion estimation for video frame interpolation," in *Proc. IEEE WACV.*, 2023, pp. 5049–5057.