2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506800

A NOVEL ALL-IN-ONE GRID NETWORK FOR VIDEO FRAME INTERPOLATION

Fanyong Xue *Jie Li** *Chentao Wu*

Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, AI Institute
Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Flow-based approaches for video frame interpolation typically consist of multiple networks that are responsible for feature extraction, optical flow estimation, and image synthesis, respectively. However, they are usually computationally expensive, and can hardly be employed in devices with limited computing resources. In this work, we propose an All-in-one Grid Frame Interpolation Network (AGFIN) to address this problem. AGFIN is a light-weight network with multiple rows and columns. In each row, we estimate the contextual features and optical flows, then the image synthesis module reconstructs the results from the warped frames and features. Each row serves as a coarser or finer auxiliary for the nearest row. In contrast to using multiple networks, our model integrates feature extraction, optical flow estimation, and image synthesis into a compact network. The experimental results show that our approach has better or comparable performance comparing to representative state-of-the-art approaches with less computational cost.

***Index Terms*—** Video frame interpolation, deep learning, video processing, optical flow estimation

1. INTRODUCTION

Video frame interpolation is a basic issue in the computer vision community and aims to synthesize plausible intermediate frames between two consecutive frames. It can be applied in video processing, like slow-motion video generation [1] and frame rate conversion [2]. Video frame interpolation can also provide an accessory for motion blur synthesis [3] and optical flow estimation [4, 5]. Long *et al*. [4] apply a generic CNN to predict the interpolation results. However, it is challenging for a generic CNN to capture motion between two frames. Niklaus *et al*. [6] propose a kernel-based paradigm, which synthesizes the in-between frames by approximating local convolution kernels for each pixel. However, kernel estimation requires a heavy computational load and it can fail to synthesize large motion beyond the kernel size. With the advances of deep learning on optical flow estimation, the flow-based paragram has been widely adopted in video frame interpolation. In general, flow-based approaches [1, 7, 8, 9] consist of feature extractors, optical flow estimators, and image synthesis models. These methods typically extract contextual features of the input frames. Then, the input frames and their contextual features are warped to the target time according to the optical flows. Finally, a synthesis network generates the results from the intermediate candidates. Although flow-based approaches have shown their potential in yielding impressive results, it is not reliable to adopt them in devices with limited computing resources.

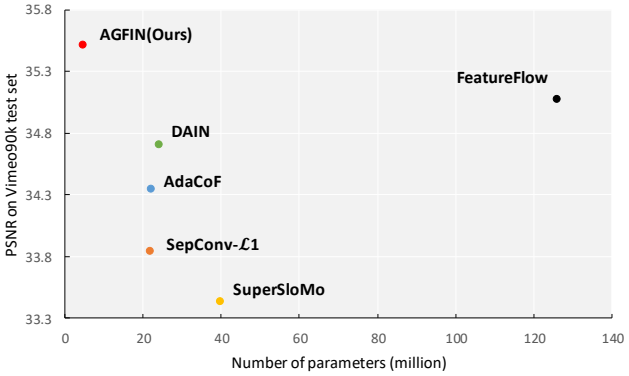
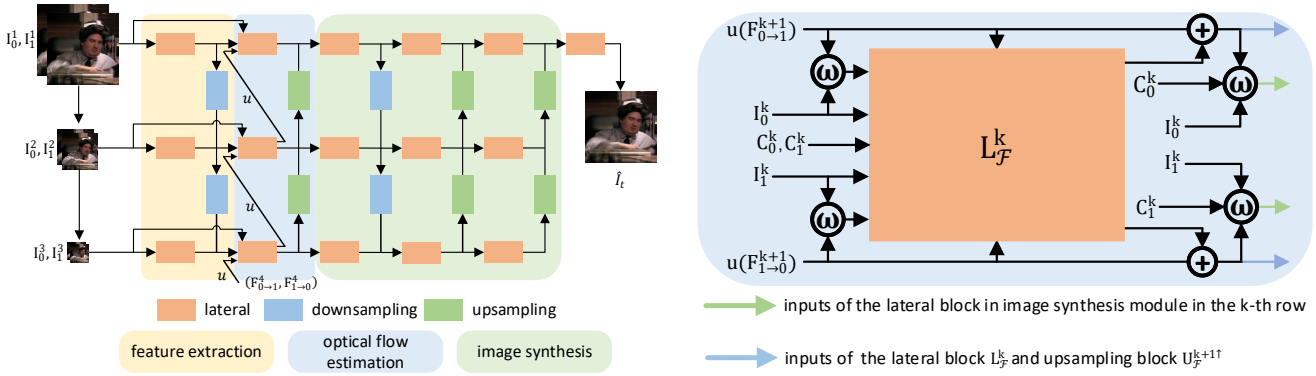


Fig. 1: **Performance and network size tradeoff of CNNs for video frame interpolation:** Our approach has the fewest parameters, and outperforms existing state-of-the-art models on the Vimeo90K dataset in terms of PSNR.

This paper represents a light-weight all-in-one grid network (AGFIN) for high-quality video frame interpolation. Instead of using integral models for feature extraction, optical flow estimation, and image synthesis, our model integrates these modules into a grid network [10]. Each module is composed of multiple rows, which can be seen as a pyramid structure. Higher rows have larger receptive fields and lower rows reconstruct results with higher quality. In each row, the input frames go through all the three modules and the outputs serve as an auxiliary for the higher or lower row using a downsampling or upsampling block, respectively. Further, our model can generate arbitrary intermediate frames in-between the in-

Authors' e-mails: {wjdncc, lijiecs}@sjtu.edu.cn, wuct@cs.sjtu.edu.cn.
*Corresponding author



(1) Framework of our proposed model with three rows (2) Flow estimation

Fig. 2: **An overview of the proposed all-in-one gird network and optical flow estimation.**

put frames. In this work, we make the following contributions: 1) we propose an AGFIN model, which significantly reduces the number of parameters by integrating different modules into a compact network; 2) we demonstrate that our proposed model has better or comparable performance comparing to the state-of-the-art video frame interpolation methods.

2. VIDEO FRAME INTERPOLATION

2.1. Overview

Given two consecutive videos frames I_0 and I_1, our goal is to generate an intermediate frame $\hat{I}_t$, where t is an arbitrary temporal location in between the two input frames. We propose a compact all-in-one gird network for video frame interpolation as illustrated in Figure 2 (1), including contextual feature extraction, optical flow estimation, and image synthesis modules. Specifically, we first downsample the input frames I_0 and I_1 to obtain K pairs frames (I_0^k, I_1^k), where K is the number of rows of our model. In the k-th row, the contextual features (C_0^k, C_1^k) are updated by learning the residual features from (I_0^k, I_1^k) and (C_0^{k-1}, C_1^{k-1}). Then, our model refines the optical flows from the coarser outputs in the $k+1$-th row. In addition, the input frames and contextual features are warped to the target time t according to the bi-directional optical flows. Finally, the image synthesis module generates the interpolation results from the warped representatives.

2.2. Video Frame Interpolation in AGFIN

Contextual Feature Extraction. Contextual features play a key role in predicting high-quality interpolation results. For $k = 2, \cdots, K$, the proposed model extracts the contextual features (C_0^k, C_1^k) as follows,

$$C_0^k = L_{\mathcal{C}}^k(I_0^k) + D_{\mathcal{C}}^{k-1\downarrow}(C_0^{k-1}) \quad (1)$$

$$C_1^k = L_{\mathcal{C}}^k(I_1^k) + D_{\mathcal{C}}^{k-1\downarrow}(C_1^{k-1}) \quad (2)$$

where $L_{\mathcal{C}}^k$ and $D_{\mathcal{C}}^{k\downarrow}$ are the lateral block and downsampling block in the feature extraction module in the k-th row, respectively. Notice that only the lateral block is used when generating C_0^1 and C_1^1.

Optical Flow Estimation. We denote the lateral and upsampling blocks in the flow estimation module in the k-th row by $L_{\mathcal{F}}^k$ and $U_{\mathcal{F}}^{k\uparrow}$, respectively. The detail of optical estimation is shown in Figure 2 (2). To capture large motion and reduce fitting complexity, we utilize backward warping with the coarser optical flows $(u(F_{1\to0}^{k+1}), u(F_{0\to1}^{k+1}))$ to narrow the distance between the corresponding pixels, where u is the bi-linear upsampling operation. Finally, the bi-directional optical flows are predicted as below,

$$(F_{0\to1}^k, F_{1\to0}^k) = L_{\mathcal{F}}^k\left(I_0^k, I_1^k, C_0^k, C_1^k, u(F_{0\to1}^{k+1}), u(F_{1\to0}^{k+1}), \omega(I_0^k, u(F_{1\to0}^{k+1})), \omega(I_1^k, u(F_{0\to1}^{k+1}))\right) + U_{\mathcal{F}}^{k+1\uparrow}(F_{0\to1}^{k+1}, F_{1\to0}^{k+1}) \quad (3)$$

where ω is the backward warping and can be implemented using bi-linear interpolation. Specifically, $(F_{0\to1}^{K+1}, F_{1\to0}^{K+1})$ are initialized with all zeros representing no displacement. Notice that $U_{\mathcal{F}}^{K+1\uparrow}$ is not considered when estimating $(F_{0\to1}^K, F_{1\to0}^K)$.

We estimate the intermediate flows $(F_{t\to0}^k, F_{t\to1}^k)$ using average flow projection layer [8], which aggregates flow vectors that pass through the same position on average. Specifically, we obtain the intermediate flows as follows,

$$F_{t\to0}^k(\boldsymbol{x}) = -t \cdot \frac{\sum_{\boldsymbol{y}\in S(\boldsymbol{x})} F_{0\to1}(\boldsymbol{y})}{\sum_{\boldsymbol{y}\in S(\boldsymbol{x})} 1} \quad (4)$$

where $S(\boldsymbol{x}) = \{\boldsymbol{y} : round(\boldsymbol{y} + t \cdot F_{0\to1}(\boldsymbol{y})), \forall \boldsymbol{y} \in I_0\}$ demonstrates all the pixels on frame I_0 that traverse the position $\boldsymbol{x}$ on frame I_t. Similarly, we could obtain $F_{t\to1}^k$ from $F_{1\to0}^k$. The optical flows and contextual features are shown in Figure 3.

Image Synthesis. To generate the interpolation results, some methods [11, 12] adaptively blend the warped frames with estimated weight maps. However, the quality of interpolation

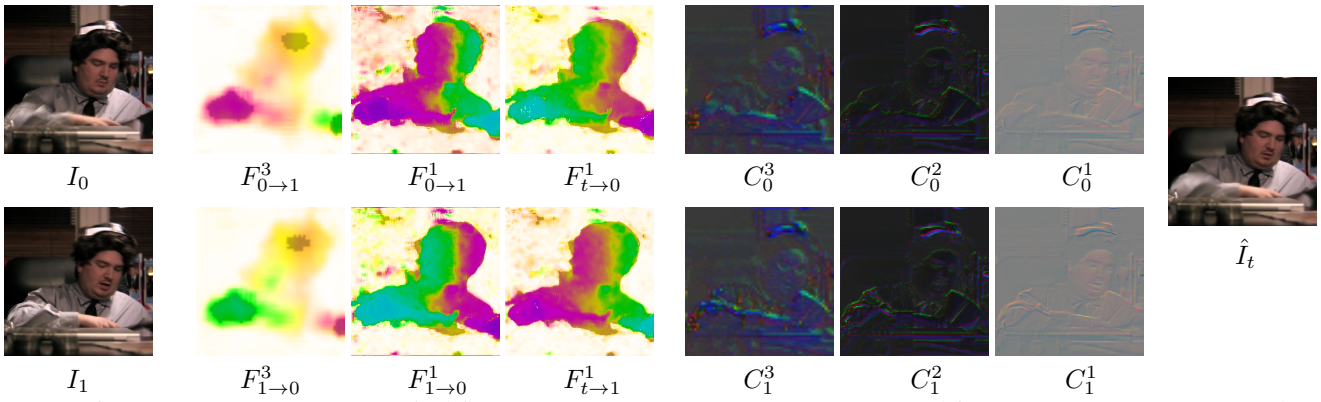


Fig. 3: **Visualization of optical flows and contextual features in different rows.** Lower rows estimate more accurate optical flows and higher rows extract contextual features with larger receptive fields.

results largely depends on that of pixel-wise correspondence between the warped frames. In this work, we use a fully convolutional module to directly synthesize the interpolation results from the warped representatives. We denote the inputs of the lateral block of the image synthesis module in the k-th row by S^k, which can be defined as follows,

$$S^k = \rho(\omega(\{I_0^k, C_0^k\}, F_{t\to0}), \omega(\{I_1^k, C_1^k\}, F_{t\to1})) \quad (5)$$

where ρ is the concatenation operation. In order to address occlusion, the first column of the image synthesis module adopts a low-to-high structure to increase receptive fields. A coarse-to-fine paradigm is used in the second and third columns for high-quality results.

Loss Function. Laplacian loss [7] has been found to perform well on quantitative benchmarks. We denote the synthesized frame by $\hat{I}_t$ and the ground truth by I_t. We extract 5 layers of Laplacian pyramid representations as follows,

$$\mathcal{L}_{Lap} = \sum_{i=1}^{5} 2^{i-1} ||L^i(\hat{I}) - L^i(I_t)||_1 \quad (6)$$

where $L^i(I)$ indicates the i-th layer of a Laplacian pyramid representation of a frame I. We employ an advanced off-the-shelf optical flow estimator as our teacher model to provide an auxiliary for our optical flow estimation module, which can be considered as a student model. Therefore, we devise an optical flow loss. By denoting the predicted results of the teacher model by $(\mathcal{F}^k_{0\to1}, \mathcal{F}^k_{1\to0})$, we define the loss as follows,

$$\mathcal{L}_{Flow} = \sum_{k=1}^{K} (||F^k_{0\to1} - \mathcal{F}^k_{0\to1}||_1 + ||F^k_{1\to0} - \mathcal{F}^k_{1\to0}||_1) \quad (7)$$

Specifically, the low resolution optical flows $\{(\mathcal{F}^k_{0\to1}, \mathcal{F}^k_{1\to0}), k \in [2, K]\}$ are obtained from $(\mathcal{F}^1_{0\to1}, \mathcal{F}^1_{1\to0})$ using bi-linear downsampling. The final total loss $\mathcal{L}$ is given by,

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{Lap} + \beta \cdot \mathcal{L}_{Flow} \quad (8)$$

In this work, we set α to 0.64 and β to 0.16, respectively.

	#Parameters (million)	PSNR	SSIM	IE
1 row	0.36	32.65	0.932	3.13
2 rows	1.04	34.65	0.958	2.39
3 rows	2.35	35.34	0.964	2.20
4 rows	4.59	35.52	0.965	2.17
ScopeFlow [13]	2.35	35.34	0.964	2.20
PWC-Net [14]	2.35	35.27	0.964	2.21

Table 1: **Ablation studies on different configurations of our proposed model on Vimeo90K.**

3. EXPERIMENTS

3.1. Dataset and Metrics

We train our model on the Vimeo90k [15] training set, including 51,312 triplets. Each frame in this training set has a resolution of 256×448 pixels. During the training process, the frames are flipped horizontally and vertically. We also reverse the temporal order of triplets to improve the generalization. The quantitative evaluation is performed on the Vimeo90K [15] and UCF101 [16, 17] test datasets. Each dataset has different resolutions and various scenarios, which ensures that our evaluation is comprehensive. We use Peak Signal-to-Noise Ratio (PSNR), Structural-Similarity-Index (SSIM), and Interpolation Error (IE) [18] as the metrics. While higher values indicate better results with the PSNR and SSIM metric, lower values indicate better results in terms of IE.

3.2. Ablation Experiments

In this section, we conduct ablation studies to account for the performance and network size tradeoff and explore the impact of different optical flow estimator teachers.

Performance and Network Size Tradeoff. We train four

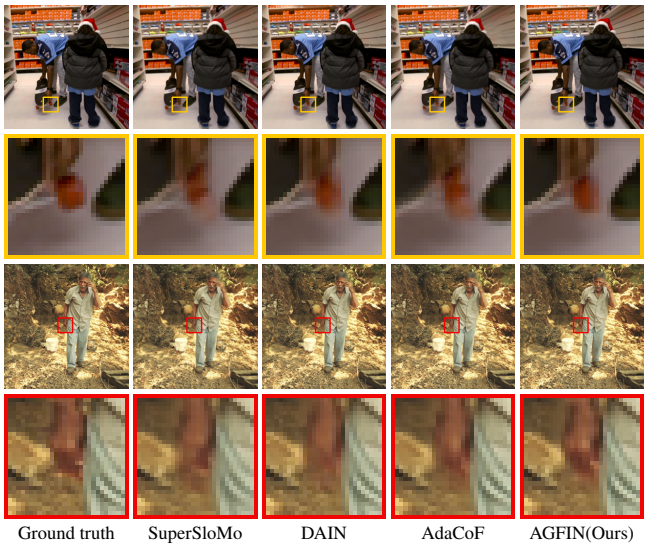


Fig. 4: **Challenging examples for video frame interpolation.** Comparing to existing video frame interpolation methods, the proposed model yields more visually pleasing results with clear boundaries.

variations with different numbers of rows. As shown in Table 1 (first section), the quality of the interpolation results improves when using more rows. In order to achieve the balance between the performance and network size, we don't try more rows. Therefore, we adopt the model with four rows in the evaluation.

Optical Flow Estimation. To analyze the effectiveness of different optical flow teachers, we train two versions of our model, one supervised by ScopeFlow [13] and one supervised by PWC-Net [14]. As shown in Table 1 (second section), both two models perform similarly well. We select ScopeFlow [13] in the evaluation.

Method	#Parameters (million)	PSNR	SSIM	IE
SuperSloMo [1]	39.61	33.44	0.950	2.73
SepConv-$\mathcal{L}_F$ [19]	21.67	33.48	0.951	2.66
SepConv-$\mathcal{L}_1$ [19]	21.67	33.85	0.955	2.49
DAIN [9]	24.03	34.71	0.964	2.23
AdaCoF [20]	21.84	34.35	0.956	2.45
FeatureFlow [21]	125.99	35.08	0.962	2.32
AGFIN (Ours)	**4.59**	**35.52**	**0.965**	**2.17**

Table 2: **Evaluation on the Vimeo90K dataset.**

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

We compare our proposed model with the representative state-of-the-art approaches, including SuperSloMo [1], SepConv [19], DAIN [9], AdaCoF [20], and FeatureFlow [21]. As shown in Table 2, our method outperforms other methods in terms of all metrics on the Vimeo90K dataset. Specifically, AGFIN gains a 0.44 dB over FeatureFlow in terms of PSNR. As shown in Table 3, we achieve a comparable performance on the UCF101 [16, 17] dataset. Unlike traditional flow-based methods [1, 9, 21] that use complete networks for contextual feature extraction, optical flow estimation, and image synthesis, the proposed all-in-one structure significantly reduces the number of parameters. Our model has better or comparable performance on given benchmarks, but the smallest model size comparing to the state-of-the-art models. Specifically, our model reduces 96.3% fewer parameters compared to FeatureFlow [21].

Method	#Parameters (million)	PSNR	SSIM	IE
SuperSloMo [1]	39.61	34.09	0.945	3.10
SepConv-$\mathcal{L}_F$ [19]	21.67	34.75	0.945	2.93
SepConv-$\mathcal{L}_1$ [19]	21.67	34.95	0.948	2.83
DAIN [9]	24.03	34.98	0.949	2.81
AdaCoF [20]	21.84	**35.16**	**0.950**	**2.76**
FeatureFlow [21]	125.99	34.91	0.949	2.92
AGFIN (Ours)	**4.59**	35.05	**0.950**	2.83

Table 3: **Evaluation on the UCF101 dataset.**

We also show the qualitative comparisons in Figure 4. These examples are subject to large motion and complex details, which are challenging scenarios for video frame interpolation. As shown in the first and second rows in Figure 4, SuperSloMo [1] and AdaCoF [20] cannot align the wheel well and produce ghosting artifacts. Although DAIN [9] captures motion of the wheel, it yields a result with blurred boundaries. In contrast, the proposed model generates a clearer result. As shown in the third and fourth rows in Figure 4, SuperSloMo [1], DAIN [9], and AdaCoF [20] all generate blurred results. In contrast, AGFIN generates a clear shape of the hand, which is closer to the ground truth. Overall, the proposed AGFIN reconstructs higher-quality interpolation results with fewer artifacts than existing state-of-the-art methods.

4. CONCLUSION

In this work, we propose an All-in-one Grid Frame Interpolation Network (AGFIN) for devices with limited computing resources. Our approach integrates contextual feature extraction, optical flow estimation, and image synthesis into a compact network. Based on the proposed all-in-one structure, our model achieves a balance between performance and network size. Further, we can use different configurations according to different applications. The quantitative and qualitative experiments demonstrate that our proposed approach has better or comparable performance comparing to existing state-of-the-art video frame interpolation methods with less computational cost.

5. REFERENCES

[1] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9000–9008.

[2] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung, "Phase-based frame interpolation for video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1410–1418.

[3] Tim Brooks and Jonathan T Barron, "Learning to synthesize motion blur," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6840–6848.

[4] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu, "Learning image matching by simply watching video," in *European Conference on Computer Vision*. Springer, 2016, pp. 434–450.

[5] Jonas Wulff and Michael J Black, "Temporal interpolation as an unsupervised pretraining task for optical flow estimation," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 567–582.

[6] Simon Niklaus, Long Mai, and Feng Liu, "Video frame interpolation via adaptive convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 670–679.

[7] Simon Niklaus and Feng Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1701–1710.

[8] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang, "Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[9] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang, "Depth-aware video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3703–3712.

[10] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf, "Residual conv-deconv grid network for semantic segmentation," *arXiv preprint arXiv:1707.07958*, 2017.

[11] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros, "View synthesis by appearance flow," in *European conference on computer vision*. Springer, 2016, pp. 286–301.

[12] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim, "Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation," in *European Conference on Computer Vision*, 2020.

[13] Aviram Bar-Haim and Lior Wolf, "Scopeflow: Dynamic scene scoping for optical flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7998–8007.

[14] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.

[15] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[16] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[17] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471.

[18] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[19] Simon Niklaus, Long Mai, and Feng Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.

[20] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee, "Adacof: Adaptive collaboration of flows for video frame interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5316–5325.

[21] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao, "Featureflow: Robust video interpolation via structure-to-texture generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14004–14013.