FeatureFlow: Robust Video Interpolation via Structure-to-texture Generation

Shurui Gui*[1,2] Chaoyue Wang*[1] Qihua Chen[1,3] Dacheng Tao[1]
[1]UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering,
The University of Sydney, Darlington, NSW 2008, Australia
[2]Department of Computer Science and Technology, University of Science and Technology of China
[3]School of Information Science and Technology, University of Science and Technology of China
agnesgsr@mail.ustc.edu.cn, chaoyue.wang@sydney.edu.au,
cqh@mail.ustc.edu.cn, dacheng.tao@sydney.edu.au

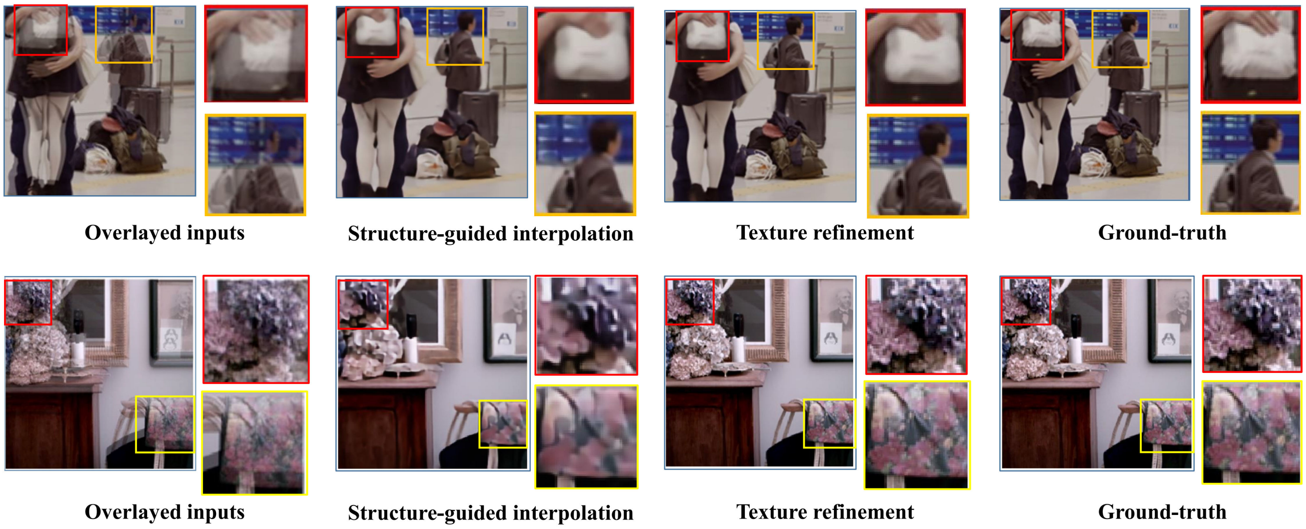

Figure 1: **Examples of the proposed structure-to-texture generation for video interpolation.** The whole framework splits the video interpolation task into two stages: *structure-guided interpolation* and *texture refinement*. The first row gives a challenging interpolation example of complicated dynamic scenes. The second row is a typical example to explain.

Abstract

Video interpolation aims to synthesize non-existent frames between two consecutive frames. Although existing optical flow based methods have achieved promising results, they still face great challenges in dealing with the interpolation of complicated dynamic scenes, which include occlusion, blur or abrupt brightness change. This is mainly because these cases may break the basic assumptions of the optical flow estimation (i.e. smoothness, consistency). In this work, we devised a novel structure-to-texture generation framework which splits the video interpolation task into two stages: structure-guided interpolation and texture refinement. In the first stage, deep structure-aware features are employed to predict feature flows from two consecutive frames to their intermediate result, and further generate the structure image of the intermediate frame. In the second stage, based on the generated coarse result, a Frame Texture Compensator is trained to fill in detailed textures. To the best of our knowledge, this is the first work that attempts to directly generate the intermediate frame through blending deep features. Experiments on both the benchmark datasets and challenging occlusion cases demonstrate the superiority of the proposed framework over the state-of-the-art methods. Codes are available on `https://github.com/CM-BF/FeatureFlow`.

1. * indicates equal contribution.

2. This work was completed during the visit of Shurui Gui and Qihua Chen (summer interns) to UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, The University of Sydney.

1. Introduction

Video frame interpolation (VFI) is an important research topic in the computer vision community. It aims to synthesize intermediate frames between any two consecutive video frames. Related techniques are wildly applied to real-world applications, such as slow-motion production [15, 2, 23, 26, 25, 19], frame rate upconversion [7, 4], video restoration [35, 33, 28].

In recent years, significant progress has been made by optical flow based methods. These methods estimate forward or backward optical flows between two original frames, and warp pixels to synthesize the intermediate frame directly. In general, the optical flow based pixel synthesis can explicitly represent the dynamic motion and reach high fidelity in the details. Moreover, recent works that adapt techniques such as bidirectional flow estimation [15], context information [23] and depth maps [2] have achieved more accurate flow estimation and better interpolation results. However, due to the basic assumptions of the optical flow estimation, *e.g.* smoothness and consistency, optical flow based methods are inherently difficult to handle the interpolation of complicated dynamic scenes which include the regions suffering from occlusion, blur or abrupt brightness change.

Deep generative models have achieved great success in a series of image/video generation tasks [42, 31, 9, 32, 10]. Comparing to existing interpolation methods, they demonstrate the potential of synthesizing highly realistic images yet would be less affected by occlusion and complex scenes. Besides, deep features demonstrate great capabilities on both extracting semantic information from visual data and synthesizing feature-aware outputs [13, 30, 38, 39]. Therefore, we believe that deep feature based generation models could be one kind of ideal solution for VFI tasks. However, since the extracted deep features would more or less lose some details, if we want to directly synthesize the intermediate frame, it will be a great challenge to keep the texture consistency between the generated frame and the original inputs.

In this paper, we devised a structure-to-texture generation framework for feature-aware video frame interpolation. The typical examples of two-stage results are shown in Figure 1. Instead of learning pixel-wise optical flow between two frames, our framework aims to explore feature flows (FeFlow) in-between corresponding deep features. Meanwhile, to solve the 'consistency' problem that may encounter by directly synthesizing, we split the interpolation progress into two stages: *structure-guided interpolation* and *texture refinement*. In the stage-I, the proposed Multi-flow Multi-attention Generator (MMG) takes two consecutive frames as inputs, and aims to predict feature flows from both of them to the intermediate frame. Besides the RGB frames, the edge images are concatenated as the 4$^{\text{th}}$ channel to reinforce structural information of the dynamic scene. Then, a coarse intermediate frame without detailed textures will be synthesized through blending deep features. In the stage-II, through aligning original frames to the coarse result generated in the stage-I, a Frame Texture Compensator (FTC) is devised to synthesize the missed texture details of the intermediate frame. The generated texture residuals are overlaid to the coarse structural result to produce the final output. Overall, we made the following contributions:

- A novel structure-to-texture generation framework is proposed for video frame interpolation tasks. To our best knowledge, this is the first work that directly generates the intermediate frame through blending deep features.
- To estimate feature flow between two frames, we devised a Multi-flow Multi-attention Generator, which divides features along the channel axis and blends them to predict target frames' features.
- To solve the inconsistency between forward/backward frames and the results produced by the generator, we developed 4$^{\text{th}}$ channel's edge inputs as structure reinforcement, and introduce triangle constraint for augmenting non-linear processing capability and alignment efficiency.
- Comprehensive experiments show that our framework can handle challenging frame interpolation cases (*e.g.* severe occlusion cases) and produces better results than state-of-the-art approaches [2, 15, 3, 25, 23, 20].

2. Related Works

Video frame interpolation (VFI) is a classic video processing task which is generally based on two steps: optical flow prediction and interpolation synthesis [2, 23, 15, 3, 20]. Meyer *et al.* [22] proposed a multi-scale pyramid model for VFI which performs impressively in cases with small motions and low-frequency contents. Then, Long *et al.* [20] made a successful attempt to use deep CNN model for optical flow generation. However, their model suffers from severe blurriness when tackles VFI. Subsequently, deep voxel flow [19] built a 3D optical flow and utilized it to warp original frames. However, even though its results contain less blurriness, it is hard for this model to handle big motions. Kernel-based methods (AdaConv [24] and SepConv [25]) estimate spatially-adaptive interpolation kernels to synthesize pixels from a large neighborhood and obtain high-quality results. But their algorithms are computationally expensive and lack of occlusion solutions.

Recently, Super SloMo [15] and CtxSyn [23] adopted visual and context maps separately to implicitly deal with occlusion problems. Moreover, DAIN [2] explicitly detects the occlusion by utilizing the depth information to manage the holes or overlay that the occlusion may cause. However, what their models do to handle the occlusion in pixel level or shallow feature layers is limited, so the results 4.3

still show their difficulties in eliminating artifacts around the boundaries due to the occlusion phenomenon. Instead, our model handles such problems in deep feature layers.

Deformable convolution. Dai *et al*. [11] first proposed the novel CNN layer - deformable convolutions (DConv), where additional calculated offsets are produced to obtain information away from its regular local kernel neighborhood. Deformable convolutions are widely used in various tasks such as video object detection [6], action recognition [21, 34], semantic segmentation [12], and video super-resolution [28, 33]. In particular, TDAN [28] and EDVR [33] use deformable convolutions to align the input frames with reference at the texture feature level without explicit pixel flow estimation and frames warping. However, we found that DConv's offsets are general many-to-one flows which could be regarded as the universal version of motion flows. DConv with its offsets together can be considered as many-to-one weighted warping. Thus, the word "flow" in this paper also represents the offset in DConv.

Attention mechanism. Attention has proven its effectiveness in many tasks [29, 36, 17, 18, 41]. Attention mechanism learns weighted maps and exerts them on inputs to imitate humans' attention mechanism, which is also a way to handle the occlusion [40]. Motivated by the success of these works, we proposed a multi attention predictor (MAP) module to cooperate with multi groups of flows. Inspired by GAN dissection [5], we assume divided features may contain objects segmentation semantics in feature level. The results show its validity.

3. Method

Given two consecutive video frames I_1 and I_2, our goal is to predict the middle frame $\tilde{I}_t$ in-between them. The proposed structure-to-texture generation framework works in two stages as illustrated in Figure 2 and 3. In stage-I, a Multi-flow Multi-attention Generator (MMG) is trained to estimate feature flows between both the input frames and the target middle one, and further synthesize the coarse interpolation result which emphasizes the overall structure. In the stage-II, we devised a Frame Texture Compensator (FTC), which aims to fill in the texture/details of the coarse result based on the original frames.

3.1. Multi-flow multi-attention generator

As shown in Figure 2, the proposed Multi-flow Multi-attention Generator (MMG) aims to align and blend two original frames in the hidden layers, then utilizes the synthetic feature to generate a coarse interpolation result. Specifically, MMG consists of three parts: feature extractor module, multi-flow multi-attention module, and global generator module. Among them, the feature extractor module extracts deep features from the input video frames. Then, the multi-flow multi-attention module is devised to explore feature flows between two consecutive frames, and further blend their features to obtain the synthetic feature of the intermediate frame. Finally, the global generator takes the synthetic feature as input and aims to generate a coarse result of the intermediate frame.

Multi-flow multi-attention feature blending. As aforementioned, the multi-flow multi-attention module takes the features F_0 and F_1 of two consecutive frames as input. Then, the multi-flow sub-module is trained to estimate feature flows, $flow_{0\to t}$ and $flow_{1\to t}$, which represent the feature flows from inputs F_0 and F_1 to the desired feature $\tilde{F}_t$, separately. Note that, in order to capture different semantic components in the video frames, the extracted features F_0 and F_1 are split into the same number of groups along the channel axis. It can be also regarded as extracting a group of features to represent different contents, such as background or objects, from each input frame. Meanwhile, we concatenate F_0 and F_1, then feed it into the Multi-Flow Predictor (MFP) to generate the same number of flow offsets for both corresponding input feature groups. Finally, utilizing the deformable convolution operation, we gain the warped features, $\hat{F}_0$ and $\hat{F}_1$.

At the second step, giving the warped features $\hat{F}_0$ and $\hat{F}_1$, we aim to learn their attention maps. With attention maps, the model could blend $\hat{F}_0$ and $\hat{F}_1$ to produce the desired synthetic feature of the intermediate video frame. Specifically, we input the concatenated feature into Multi-Attention Predictor (MAP) to create $2 \times n$ attention maps A_0 and A_1 for features $\hat{F}_0$ and $\hat{F}_1$, respectively. Here, the number of attention maps, *i.e.* n, is equal to the number of feature flows mentioned above. Finally, based on the learned attention maps, the synthetic feature $\tilde{F}_t$ is the weighted combination of the results of all groups.

We adopted blending loss to optimize the study of $\tilde{F}_t$. Ground-truth frame I_t^{gt} is fed into *feature extractor* to produce the ground-truth features, noted as F_t^{gt}. Thus, the blending loss is defined as:

$$\mathcal{L}_b = \rho(\tilde{F}_t - F_t^{gt}), \quad (1)$$

where $\rho(diff) = \sqrt{diff^2 + \epsilon^2}$ is Charbonnier penalty function [8]. ϵ is generally $1e-6$.

Structure-guided generation. We argue that structure information such as edge is significant for the subsequent texture refinement (*i.e.* stage-II). Because of concentrating on learning feature flows between the deep features, in the stage-I, the generated intermediate frame may not contain enough texture details. It will cause great challenges for the feature alignment and texture refinement in the stage-II. Therefore, we hope our model could pay more attention to the lines and edges of both the input and generated frames.

Therefore, we adopted BDCN [14] to generate edge images from original frames, noted as E_0 and E_1. After concatenating them with I_0 and I_1, we input these combina-

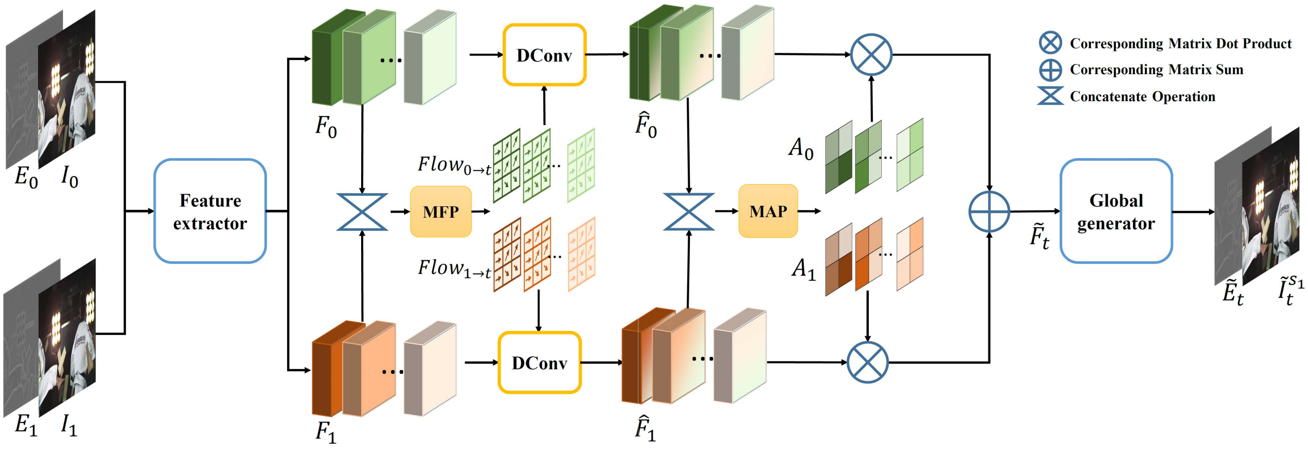

Figure 2: **Stage-I: structure-guided Multi-flow Multi-attention Generator (MMG).** Given two original frames and their detected edges, the proposed multi-flow predictor (MFP) calculates feature flows between original frames and the intermediate frame by using extracted features. Then, deformable convolutions (DConv) are adopted to produce warped features. Subsequently, we use warped features to calculate multi-attention maps through the multi-attention predictor (MAP) module. Afterwards, we calculate dot products of $2 \times n$ attention maps and corresponding $2 \times n$ groups of warped features. Then we merge the two attention weighted features to synthesize the intermediate feature. Finally, the synthetic feature is utilized to generate the output intermediate frame and its edge image.

tions to *feature extractor* to produce F_0 and F_1. Passing the *multi-flow multi-attention* and *global generator* modules, the coarse interpolation results $\tilde{I}_t^{s_1}$ and their edge images $\tilde{E}_t$ are synthesized. We set the edge loss as:

$$\mathcal{L}_e = ||\tilde{E}_t - \mathcal{E}(I_t^{gt})||_2^2, \tag{2}$$

where $\mathcal{E}$ is the BDCN network for extracting the edge image of I_t^{gt}. In this stage, edge images act as structure guidance. They enforce structure ingredients in deep feature layers for blending. Through generating the edge image $\tilde{E}_t$, we hope that the structure information will not be lost during the stage-I. Consequently, the generated coarse result $\tilde{I}_t^{s_1}$ is forced to gain more structure information which benefits subsequent texture refinement.

Triangle constraint. Instead of predicting flows $flow_{0\to 1}$, $flow_{1\to 0}$ and assuming that the motion is always linear as most existing works, we attempt to produce the motion flows from the middle feature to input features directly: $flow_{0\to t}$ and $flow_{1\to t}$.

Linear motion representation may fail when occlusion occurs. Under the assumptions of flow estimation, in the occlusion case, pixels in different places may merge into one location. Since we warp in deep feature layers, it is possible to use the feature information to predict nonlinear result locally due to the occlusion problem, rather than use pre-defined rules [2]. In addition, the model's capability to handle predictable nonlinear motion such as deformation will be strengthened, as shown in 4.3.

Without constraints, predicted $flow_{0\to t}$ and $flow_{1\to t}$ may warp F_0 and F_1 to different location, which is counterintuitive and has negative impact on the following attention prediction. We used triangle constraint that requires motion vectors' heads to locate in the same place to align two warped features and synthesize higher quality $\tilde{F}_t$. Accordingly, we set a triangle loss before attention operations, which can be described as following:

$$\mathcal{L}_{tri} = \rho(\hat{F}_0 - F_t^{gt}) + \rho(\hat{F}_1 - F_t^{gt}), \tag{3}$$

where $\hat{F}_0$ and $\hat{F}_1$ represent the features that warped by the offsets predicted by the MFP, F_t^{gt} has the same meaning as mentioned in $\mathcal{L}_b$ (Eq.1).

Finally, for overall color and frame synthesis guidance, we used pixel-wise loss as follow:

$$\mathcal{L}_g = \sigma * \rho(\tilde{I}_t^{s_1} - I_t^{gt}), \tag{4}$$

where $\sigma = 128$ which represents the inverse normalization.

3.2. Frame texture compensator

In the second stage, giving the generated coarse result, a Frame Texture Compensator (FTC) is trained to fill in the missed texture details. As shown in Figure 3, inspired by TDAN [28] and EDVR [33], our FTC aims to locate and borrow the desired texture features from the original input frames. Specifically, after extracting features from the input frames, the Pyramid Cascading and Deformable convolutions (PCD) align module aligns the original frames I_0 and I_1 with the reference frames $\tilde{I}_t^{s_1}$. Then, unitizing both the Temporal and Spatial Attention (TSA) fusion module and texture generator, the network compares the texture

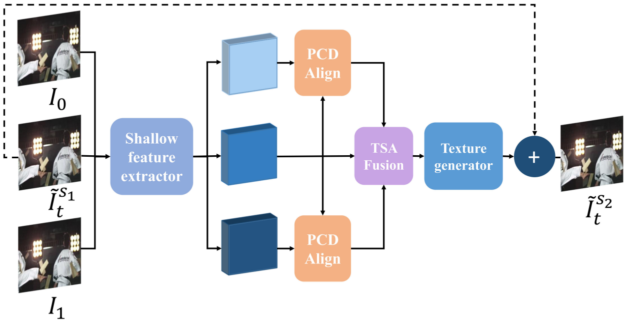

Figure 3: **Architecture of proposed stage-II: Frame Texture Compensator (FTC).** Given two original frames and a result frame from stage-I, we use shallow feature extractor to gain their texture level features. Then we adopt PCD module [33] to align the both-side features with the middle one with the help of the TSA Fusion module and feed the result to texture generator. Finally, we overlay the residue which is produced by texture generator on the stage-I interpolation result to generate the final output frame.

differences between original frames and $\tilde{I}_t^{s_1}$ to synthesize residues. Finally, we overlap the learned residues to $\tilde{I}_t^{s_1}$ to gain the final high-quality interpolation results $\tilde{I}_t^{s_2}$. The reconstruction loss is defined as:

$$\mathcal{L}_r = \sigma * \rho(\tilde{I}_t^{s_2} - I_t^{gt}), \tag{5}$$

where the σ is the same as the one in $\mathcal{L}_g$ (Eq. 4).

In the proposed FTC, PCD and TSA modules are adopted from the EDVR [33] framework, because they demonstrate the strong ability of feature alignment and fusion. However, different from video restoration tasks, in our case, the original and reference frames haven't got the same image qualities. On the one hand, it asks us to adjust the network architectures, such as remove downsampling or upsampling layers, for reasonable performance. On the other hand, the alignment requirement inspired us the structure-guidance strategy in the stage-I, which contributes to training robust and final performance. Since the space limitation, more implementation and architecture details can be found in our supplementary materials.

3.3. Implementation details

MMG loss functions. The total loss function of MMG can be shown as follow:

$$\mathcal{L}_{MMG} = \mathcal{L}_g + \lambda_b * \mathcal{L}_b + \lambda_{tri} * \mathcal{L}_{tri} + \lambda_e * \mathcal{L}_e \tag{6}$$

where $\lambda_b = 500$, $\lambda_{tri} = 20$ and $\lambda_e = 5$. A grid search was performed to determine these hyper-parameters. In addition, the empirical studies suggest the proposed model is robust to our losses.

FTC loss functions. In the stage-II, we simply adopted a reconstruction loss to optimize the final interpolation result, *i.e.* $\mathcal{L}_{FTC} = \mathcal{L}_r$.

Training strategy. We used the Adam [16] to optimize the proposed network where we set the β_1 and β_2 to 0.9 and 0.999. We trained our model in two stages. In the first step, we used a batch size of 16 and the initial learning rates were set to $1e^{-4}$. The learning rates were reduced by a factor of 0.25 after training for every 30 epochs. We trained MMG (section 3.1) in this step for 65 epochs. In the second step, we used a batch size of 4 and the initial learning rates were set to $1e^{-4}$. We trained the FTC (section 3.2) for 20 epochs then reduced its learning rates by a factor of 0.25. After keeping training for another 7 epochs, we reduced its learning rates the second time and stopped after 2 epochs. We trained our model on two RTX 2080Ti GPU cards, which took about 5 days to converge.

4. Experiments

In this section, we first introduce the evaluation datasets and metrics in our experiments. Then, we conduct ablation study to analyze the contribution of the proposed edge loss, triangle loss, and multi-flow multi-attention module. Moreover, we quantitatively and qualitatively compare our model with state-of-the-art video frame interpolation methods.

4.1. Datasets and evaluation metrics

Training dataset. We used the Vimeo90K dataset [37] to train our model. The Vimeo90K dataset has 51,312 triplets for training, where each triplet contains 3 consecutive video frames with a resolution of 256 x 448 pixels. We train our network to predict the middle frame. We performed data augmentation by horizontal flipping as well as reversing the temporal order of the triplet.

Test datasets. In this paper, our model was trained on a single training set but validated on different test sets, which include different resolutions, scenes, shooting equipment, and anime. Specifically,

- **Vimeo90K test set.** Vimeo90K [37] contains 3,782 triplets in its test set for VFI evaluation, where all of the images with the resolution of 448×256 pixels.
- **Adobe240-fps.** Adobe-240fps [27] is a set of real world videos which was originally used as Video Deblur. Following [15], we extracted 10% of it and transfer it to 622 triplets with the resolution of 640×360.
- **Middlebury.** The Middlebury benchmark [1] is widely used to evaluate VFI methods. We evaluated our model on its EVALUATION set by uploading the results to the benchmark website, where groundtruth is hidden. The image resolution in it is around 640×480.

	Vimeo90K		Adobe240fps		Occ	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
FeFlow-None	34.16	0.9714	32.52	0.9542	36.32	0.9809
FeFlow-Edge	34.84	0.9744	32.54	0.9542	36.80	0.9813
FeFlow-Full	**35.28**	**0.9764**	**32.66**	**0.9550**	**37.12**	**0.9826**

Table 1: Effect of edge and triangle loss functions

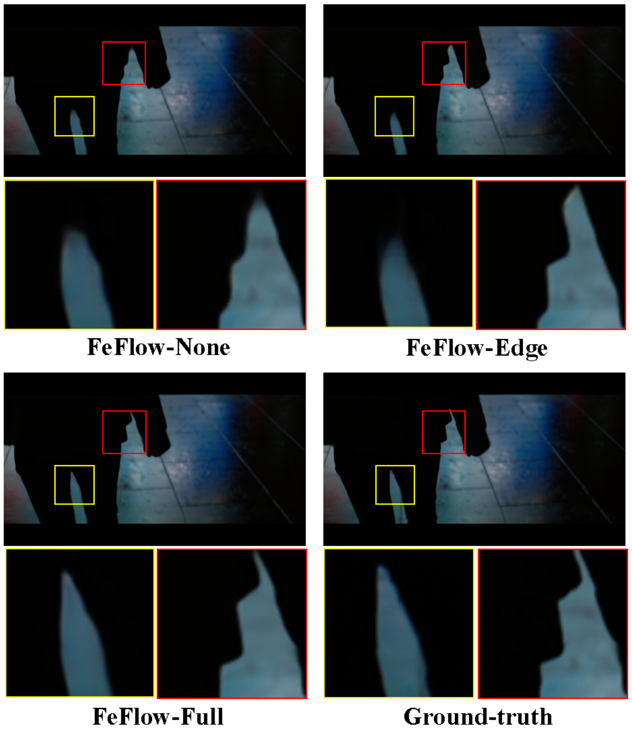

Figure 4: **Effect of edge and triangle loss functions.**

- **Occ.** Since occlusion is one of the most challenging cases for existing video interpolation methods, we collected 29 triples from YouTube videos which specifically attempts to contain obvious occlusion conditions. The resolution of select frames are 640 × 360 pixels.

Metrics. Generally, we evaluate models by measuring PSNR as well as SSIM. In Middlebury EVALUATION set, Middlebury benchmark evaluates models in terms of interpolation error on *disc.*, *i.e.* regions with discontinuous motion, and *unt.*, *i.e.* textureless regions.

4.2. Ablation study

Loss functions. Several loss functions are proposed in our framework. Among them, edge loss is devised to enhance the structure information input to the feature layers and aim to achieve results with clearer edges. Triangle loss is devised to further align warped features for the subsequent multi-attention module. To analyze the effectiveness of these losses, we performed the following variations:

	Vimeo90K		Adobe240fps		Occ	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1 group	34.17	0.9710	31.41	0.9474	35.25	0.9773
4 groups	35.02	0.9755	32.54	0.9542	37.08	0.9825
16 groups	**35.28**	**0.9764**	**32.66**	**0.9550**	**37.12**	**0.9826**

Table 2: Effect of different number of groups of attention maps

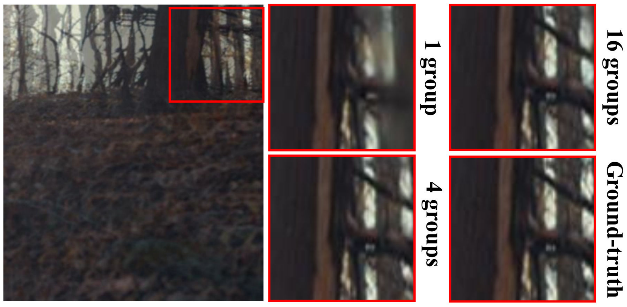

Figure 5: **Effect of the number of groups of multi-flow multi-attention.** 1 group model often fails at the boundary of the image because it is the place that the occlusion generally happens. Note that this occlusion is caused by the image boundary.

- **FeFlow-None.** We removed both the BDCN edge prediction model (*i.e.* edge loss) and the triangle alignment loss. Note that, in this variant, the input and output are set to 3 channels RGB images.
- **FeFlow-Edge.** We only eliminated the triangle loss from the original model where the edge-channel and edge loss are utilized in this variant.
- **FeFlow-Full.** The model was trained using all loss functions that devised in this work.

According to the quantitative comparison (Table 1), FeFlow-Full achieved the best performance over all test data. Meanwhile, FeFlow-Edge is better than FeFlow-None in most metrics. As shown in Figure 4, the FeFlow-None almost failed to recover the edge shape of the cloth. The FeFlow-Edge model benefits from learned edge information, thus, the cloth with triangle shapes was synthesized and the edges are clearer. However, without triangle loss, the warped features may unable to be aligned well, which leads to space shifts. Instead, the FeFlow-Full handles this problem well and achieved the best result in tackling the blurriness between legs.

Multi-flow multi-attention module. To analyze the significance of the multi-flow multi-attention module, we trained variations of 1, 4 and 16 groups of flows and attention maps. As shown in Table 2, as the number of groups increases, the interpolation results become better. It is mainly because different channel groups could focus on different kinds of motion/content. In Figure 5, occlusion problems

Method	Average			Mequon			Schefflera			Urban			Teddy			Backyard			Basketball			Dumptruck			Evergreen		
	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.
SepConv-L_1 [25]	5.61	8.74	2.33	2.52	4.83	1.11	3.56	5.04	1.90	4.17	4.15	2.86	5.41	6.81	3.88	10.2	12.8	3.37	5.47	10.4	2.21	6.88	15.6	1.72	6.63	10.3	1.62
ToFlow [37]	5.49	8.55	2.17	2.54	4.35	1.16	3.70	5.19	1.88	3.43	3.89	1.93	5.05	6.43	3.39	9.84	12.3	3.42	5.34	10.0	2.28	6.88	15.2	1.61	7.14	11.0	1.69
Super SloMo [15]	5.31	8.39	2.12	2.51	4.32	1.25	3.66	5.06	1.93	2.91	4.00	**1.41**	5.05	6.27	3.66	9.56	11.9	3.30	5.37	10.2	2.24	6.69	15.0	1.53	6.73	10.4	1.66
CtxSyn [23]	5.28	8.00	2.19	**2.24**	**3.72**	**1.04**	**2.96**	**4.16**	**1.35**	4.32	3.42	3.18	**4.21**	**5.46**	**3.00**	9.6	11.9	3.46	5.22	9.8	2.22	7.02	15.4	1.58	6.66	10.2	1.69
MEMC-Net [3]	5.00	7.71	2.20	2.39	3.92	1.28	3.36	4.52	2.07	3.37	3.86	2.20	4.84	5.93	3.72	8.55	10.6	3.14	**4.70**	**8.81**	**2.03**	6.40	14.2	1.58	6.37	9.87	1.57
DAIN [2]	4.86	7.61	**2.08**	2.38	4.05	1.26	3.28	4.53	1.79	3.32	3.77	2.05	4.65	5.88	3.41	7.88	9.74	**3.04**	4.73	8.90	2.04	6.36	14.3	**1.51**	**6.25**	**9.68**	**1.54**
FeFlow	**4.82**	**7.41**	2.12	2.28	3.73	1.18	3.50	4.78	2.09	**2.82**	**3.13**	1.66	4.75	5.78	3.72	**7.62**	**9.40**	**3.04**	4.74	8.88	**2.03**	**6.07**	**13.1**	1.59	6.78	10.5	1.65

Table 3: Evaluation on Middlebury benchmark. disc.: regions with discontinuous motion, and unt.: textureless regions. The numbers in **boldface** and blue represent the best and second best performance.

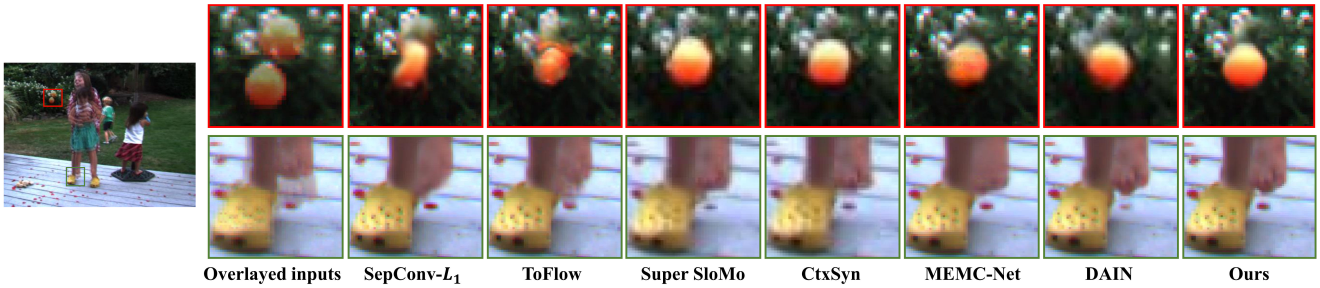
Overlayed inputs **SepConv-L_1** **ToFlow** **Super SloMo** **CtxSyn** **MEMC-Net** **DAIN** **Ours**

Figure 6: **Visualized examples on Middlebury EVALUATION set.** FeFlow shows its strong capacity on dealing occlusion and semantic shape distortions by generating high quality details on balls, white flowers, rose petals, foot and slippers.

Method	Vimeo90K		Adobe240fps		Occ	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MIND [20]	33.50	0.9429	-	-	-	-
DVF [19]	31.54	0.9426	-	-	-	-
ToFlow [37]	33.73	0.9682	-	-	-	-
SepConv-L_f [25]	33.45	0.9674	31.93	0.9492	36.26	0.9804
SepConv-L_1 [25]	33.79	0.9702	32.08	0.9512	36.57	0.9816
MEMC-Net [3]	34.40	0.9743	32.42	0.9537	36.79	0.9819
DAIN [2]	34.71	0.9756	32.51	0.9539	36.98	0.9825
FeFlow	**35.28**	**0.9764**	**32.66**	**0.9550**	**37.12**	**0.9826**

Table 4: Evaluation on the Vimeo90K, Adobe240fps and Occ datasets.

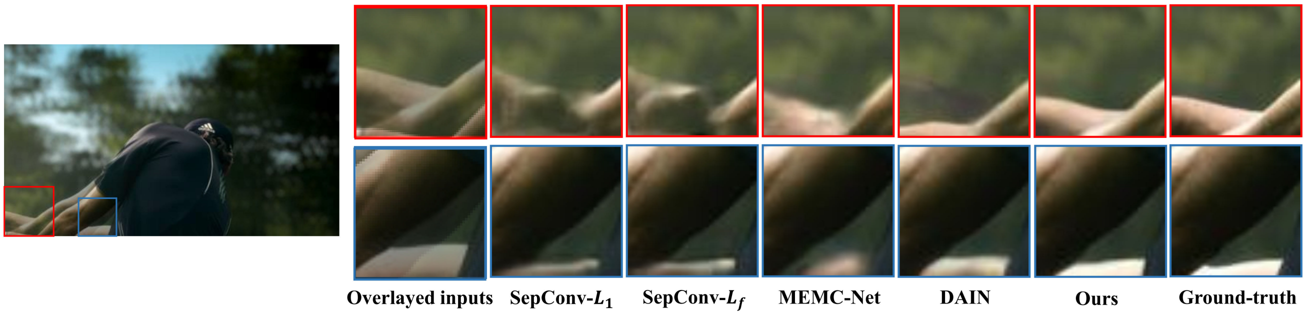
Overlayed inputs **SepConv-L_1** **SepConv-L_f** **MEMC-Net** **DAIN** **Ours** **Ground-truth**

Figure 7: **Visualized examples on Vimeo90K test set.** These examples have obvious occlusion phenomena, so the visualized results on them show these methods' abilities on handle such problems.

are alleviated when there are more groups of flows and attention maps. When there is only 1 flow-attention group, the training process is unstable, and the result shows that it encounters great challenges in dealing with occlusion. When the number of groups reaches four, the model shows its potential to handle occlusion and the overall results become better. The setting with 16 groups achieved the best result, which demonstrated strong capability in tackling with occlusion. After 16 groups, the performance increases slowly. Considering the balance between the performance and computation cost, we finally selected the setting with 16 groups.

4.3. Comparisons with state-of-the-arts

In this section, we evaluated our FeFlow model against the following VFI algorithms: MIND [20], DVF [19], ToFlow [37], Sepconv [25], Super SloMo [15], CtxSyn [23], MEMC-Net [3] and DAIN [2].

First, we compared these models on the Middlebury EVALUATION set, where we uploaded our model's results and got other state-of-the-arts results' indexes on its website[1]. As shown in Table 3, our model performs favorably against all the comparisons. The visualization comparisons in Figure 6 show that our model has its unique merit in dealing with objects' shape changes and occlusions against other methods. To be specific, ToFlow, Sepconv and MEMC-Net cannot handle the big movement of the orange ball. In the upper patch, one of the balls in overlaid inputs is circle while the other's top part is a triangle, where CtxSyn and DAIN obviously fail to comprehend this detail. Specially, DAIN's ball is smaller than the original one, and the white flowers behind are forced to be blurry because of the occlusion. In the lower patch, except DAIN, most of them are hard to tackle the holes of the slippers and the movements of the foot. Nonetheless, half of the red petal blow the foot disappears in DAIN's results. In contrast, our model solves these problems perfectly.

In Table 4, we made quantitative comparisons with state-of-the-art methods [20, 19, 37, 25, 3, 2] on several test datasets. Our approach processes them better than all the existing methods. For example, on the Vimeo90K, FeFlow has 0.57dB gain over DAIN in terms of PSNR. Figure 7 shows an example in the Vimeo90K test set, our proposed MMG exerts its merit to promise results' semantic correctness, and the FTC produce the results with high quality, while all of other algorithms fail to produce the forearm.

Considering occlusion is one of the most challenging situations in the VFI tasks, we further visually compare our method with existing works in our *Occ.* test set. As shown in Figure 8, facing the occlusion caused by the caption, our method offers the best solution. In addition, in Figure 9, we attempt to interpolate a challenging case raised by large object motion. According to the result, although the proposed FeFlow fails to make it perfect, it tries the best to generate a semantically correct result and performs better than the state-of-the-art optical-flow based method.

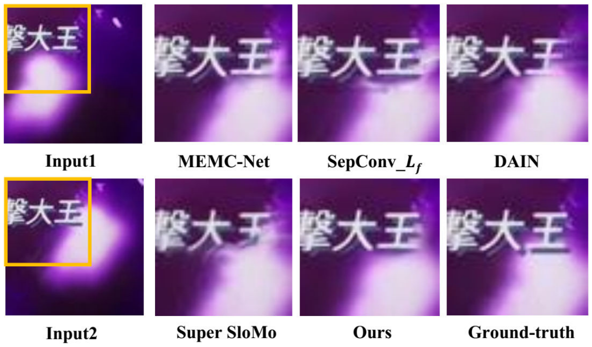

Figure 8: **Visualized examples on an occlusion case.** One of the most common scenes of occlusion is the occlusion caused by caption.

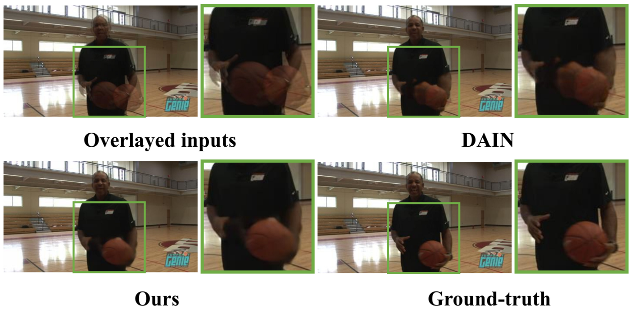

Figure 9: **A example of challenging cases.** DAIN's result is the best among compared algorithms except ours.

Worth to mention that on the Vimeo90K test set, FTC gained the improvement of 0.0499 on SSIM and 4.29dB on PSNR while the quality of the result of stage-I will decide the improvement of the frame in stage-II.

5. Conclusion

In this work, we devised a novel feature flow based structure-to-texture VFI generation algorithm for high-quality results. To our best knowledge, this is the first work that attempts to directly generate the intermediate frame through blending deep features. We exploited the efficiency of the edge and triangle loss. The proposed algorithm is efficient and accurate. Extensive quantitative and qualitative evaluations demonstrate that the proposed method performs favorably against existing frame interpolation algorithms on diverse datasets, especially in severe occlusion cases. In future works, we hope to further explore the usage of semantic information in VFI problems and dig out the relationship between it and other applications.

Acknowledgment

This work was supported by Australian Research Council Project FL-170100117.

[1] http://vision.middlebury.edu/flow/eval/results/results-i1.php

References

[1] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92(1):1–31, Mar 2011.

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] 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*, 2018.

[4] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao. High-order model and dynamic filtering for frame rate up-conversion. 27(8):3813–3826, Aug 2018.

[5] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Zhou Bolei, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

[6] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[7] R. Castagno, P. Haavisto, and G. Ramponi. A method for motion adaptive frame rate up-conversion. *IEEE Transactions on Circuits and Systems for Video Technology(TCSVT)*, 6:436 – 446, 1996.

[8] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168–172 vol.2, Nov 1994.

[9] Xinyuan Chen, Chang Xu, Xiaokang Yang, Li Song, and Dacheng Tao. Gated-gan: Adversarial gated networks for multi-collection style transfer. *IEEE Transactions on Image Processing*, 28(2):546–560, 2018.

[10] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 164–180, 2018.

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[12] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang. Restricted deformable convolution based road scene semantic segmentation using surround view cameras. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[13] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10542–10552. Curran Associates, Inc., 2019.

[14] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[17] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[18] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1673–1682. Curran Associates, Inc., 2018.

[19] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[20] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu. Learning image matching by simply watching video. In *The European Conference on Computer Vision (ECCV)*, 2016.

[21] Khoi-Nguyen C. Mac, Dhiraj Joshi, Raymond A. Yeh, Jinjun Xiong, Rogerio S. Feris, and Minh N. Do. Learning motion in feature space: Locally-consistent deformable convolution networks for fine-grained action detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[22] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[24] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[25] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[26] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally deformable alignment network for video super-resolution. *arXiv preprint arXiv:1812.02898*, 2018.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc., 2017.

[30] Chaoyue Wang, Chaohui Wang, Chang Xu, and Dacheng Tao. Tag disentangled generative adversarial network for object image re-rendering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2901–2907, 2017.

[31] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing*, 27(8):4066–4079, 2018.

[32] Chaoyue Wang, Chang Xu, Xin Yao, and Dacheng Tao. Evolutionary generative adversarial networks. *IEEE Transactions on Evolutionary Computation*, 23(6):921–934, 2019.

[33] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[34] Junwu Weng, Mengyuan Liu, Xudong Jiang, and Junsong Yuan. Deformable pose traversal convolution for 3d action and gesture recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[35] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. Optical flow guided tv-l1 video interpolation and restoration. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2011.

[36] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[37] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

[38] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[39] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017.

[40] S. Zhang, J. Yang, and B. Schiele. Occluded pedestrian detection through guided attention in cnns. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6995–7003, June 2018.

[41] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.