MSEConv: A Unified Warping Framework for Video Frame Interpolation

XIANGLING DING, Hunan University of Science and Technology, Xiangtan, China
PU HUANG, Changsha University of Science and Technology, Changsha, China
DENGYONG ZHANG*, Changsha University of Science and Technology, Changsha, China
WEI LIANG, Hunan University of Science and Technology, Hunan Key Laboratory for Service computing and Novel Software Technology, Xiangtan, China
FENG LI, Changsha University of Science and Technology, Changsha, China
GAOBO YANG, Hunan University, Changsha, China
XIN LIAO, Hunan University, Changsha, China
YUE LI, University of South China, Hengyang, China

Within the context of video frame interpolation, complex motion modeling is the task of capturing, in a video sequence, where the moving objects are located in the interpolated frame, and how to maintain the temporal consistency of motion. Existing video frame interpolation methods typically assign either a fixed size of the motion kernel or a refined optical flow to model complex motions. However, they have the limitation of data redundancy and inaccuracy representation of motion. This paper introduces a unified warping framework, named multi-scale expandable deformable convolution (MSEConv), for simultaneously performing complex motion modeling and frame interpolation. In the proposed framework, a deep fully convolutional neural network with global attention is proposed to estimate multiple small-scale kernel weights with different expansion degrees and adaptive weight allocation for each pixel synthesis. Moreover, most of the kernel-based interpolation methods can be treated as the special case of the proposed MSEConv, thus, MSEConv can be easily transferred to other kernel-based frame interpolation methods for performance improvement. To further improve the robustness of motion occlusions, an operation of mask occlusion is introduced. As a consequence, our proposed MSEConv shows strong performance on par or even better than the state-of-the-art kernel-based frame interpolation works on public datasets. Our source code and visual comparable results are available at https://github.com/Pumpkin123709/MSEConv.

CCS Concepts: • **Computing methodologies** → **Computer vision problems**; • **Reconstruction**;

Additional Key Words and Phrases: Video Frame Interpolation, Unified Framework, Complex motion Modeling, Mask Occlusion

*Correspondence author

Authors' addresses: Xiangling Ding, xianglingding@hnust.edu.cn, Hunan University of Science and Technology, Xiangtan, China; Pu Huang, pumpkin@stu.csust.edu.cn, Changsha University of Science and Technology, Changsha, China; Dengyong Zhang, zhdy@csust.edu.cn, Changsha University of Science and Technology, Changsha, China; Wei Liang, wliang@hnust.edu.cn, Hunan University of Science and Technology, and Hunan Key Laboratory for Service computing and Novel Software Technology, Xiangtan, China; Feng Li, lif@csust.edu.cn, Changsha University of Science and Technology, Changsha, China; Gaobo Yang, yanggaobo@hnu.edu.cn, Hunan University, Changsha, China; Xin Liao, xinliao@hnu.edu.cn, Hunan University, Changsha, China; Yue Li, liyue@usc.edu.cn, University of South China, Hengyang, China.

ACM 2375-4699/2024/2-ART
https://doi.org/10.1145/3648364

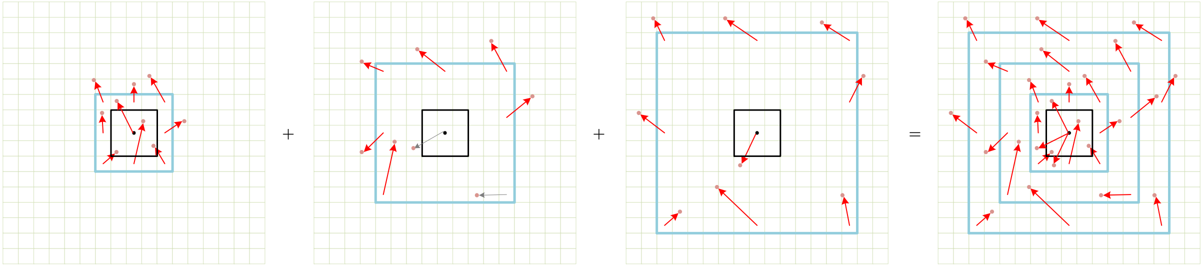
1 INTRODUCTION

Video frame interpolation (VFI) is a longstanding research task in the video processing field, which aims to synthesize non-existent intermediate frames between two consecutive video frames. Its goal is to improve the temporal super-resolution of the video sequence and maintain the temporal consistency of motion. It compensates for the motion information and then interpolates missing details for achieving a better visual appearance. As a result, VFI can effectively enhance video quality to meet people's video-viewing needs. It plays an important role in wide applications such as frame rate conversion [3], slow motion generation [2, 20], video compression, video restoration, intra-prediction in video coding and novel view synthesis [14, 55]. Though fundamental, the issue is also challenging in that the occlusion, complex motion, and feature change in real-world videos are hard to represent in a precise way.

The existing VFI methods can be roughly classified as the traditional VFI algorithms, and the convolution operation-based VFI approaches. Traditional VFI algorithms generally adopt a two-step strategy: first estimates motion information, typically optical flow, and then warp the input frame to synthesize pixels to be interpolated. This category of VFI algorithms heavily relies on the accuracy of the estimated motion information. Therefore, due to the inaccuracy of motion information, the synthesized in-between frames usually occur blurring or ghost effects in some of the challenging situations, e.g. the large motion and occlusion.

To make the interpolated results more robust in challenging situations, another major trend in this domain uses the convolution operation to replace such a two-step strategy [6, 25, 35, 36]. Specifically, they first estimate either a regular [35, 36] or an irregular [5, 6, 9, 25, 52] local kernel for each pixel of the input frames through a deep convolutional neural network (DNN), followed by sampling the surrounding related effective pixels with the estimated kernel. Therefore, in this kind of VFI method, the quality and efficiency of the interpolated frame directly depend on the region of sampling and the number of related pixels. For the regular local kernel-based methods utilize a pair of 2D kernels [35] or four 1D kernels [36] to sample pixels in the input frame. But, they cannot handle motions that exceed the kernel size range, and thus generate too many associated pixels causing large storage and computing requirements. In contrast, the irregular local kernel-based approaches [5, 6, 9, 25, 52] increase the offset vector into the regular local kernel-based methods to make the estimated local convolution shape more suitable for real motion. However, the offset vector is not sufficient to characterize the large motion and occlusion. Although these two types of VFI methods can produce better-interpolated results for the large motion by simply and roughly expanding kernel size, there are still a few limitations: (1) Capturing larger motions by expanding kernel size can produce severe data redundancy. In this case, each pixel in the in-between frame covers more pixels in the input frames by the estimated local kernel causing a visually visible ghosting effect. (2) The number of the motion magnitude for the large motion exceeds the estimated kernel size or maximum offset causing them not to be modeled correctly. Under these circumstances, the interpolated frames inevitably appear blurring in the motion regions for imprecise motion information.

To deal with the above limitations, we introduce a unified warping framework (MSEConv) that exploits multi-scale expandable deformable convolution to estimate multiple small-scale kernel weights with different expansion degrees and adaptive weight allocation for the synthesis of each pixel. In this paper, multiple small kernels with different expansions are first estimated for sampling the input frames. Then the target pixel is produced by linearly warping the sample values of the input frame to the target pixels. Compared to existing methods [6, 9, 25, 35, 36, 52], we first use multiple small kernels instead of simply expanding kernel size to reduce the growth rate of the estimated number of pixels. For the large kernel, the number of sampling for each pixel is $(N \times N)$, while the proposed MSEConv reduces it to $n \times n \times G$, where G is the group number of kernel and $G, n \ll N$. Secondly, we use different expansion steps for each kernel to break through the limitation of insufficient offset vector and reduce resampling. Third, our method has the highest sampling efficiency, thus, the proposed MSEConv can handle the complex motion in a real-world video. The sample location of the target pixel



(a) d=1 (b) d=3 (c) d=5 (d) MSEConv

Fig. 1. The sample location (light blue box) of the target pixel (black box) by estimated kernel size with different expansion for sampling, and dilation coefficient d (a) 1, (b) 3, and (c) 5, respectively. (d) The sample location of the synthesis target pixel by MESCov for sampling with dilation coefficient $d = 1, 3$, and 5. When the kernel is fixed as $1, 3$, or 5, it is refereed as the AdaCoF [25].

is displayed in Figure 1. From this figure, we can observe that the proposed MSEConv can efficiently represent the motion with more delicate information on the sample location, even in the case of a large motion. Furthermore, we strengthen the expression of global features by introducing a global feature extraction module. To further enhance the network's ability to handle occlusion and more robustness to motion occlusion, we randomly occluded the part of the motion regions of the input frames in the training dataset. The proposed MSEConv is trained in an end-to-end way. Both quantitative and qualitative experiments indicate that our framework MSEConv displays strong performance on par or even better than the state-of-the-art kernel-based VFI methods.

The contributions of this paper are summarized as follows:

- A novel unified warping framework MSEConv for frame interpolation is proposed. It not only learns multiple small kernels, but also assigns different expansions and allocates adaptive weight to each kernel. Thus, it can use fewer estimated pixels to process more large motions.
- Most of the kernel-based and flow-based VFI methods can be treated as the special case of the proposed MSEConv.
- A mask occlusion training strategy is introduced to enhance the robustness of the network to handle the occlusion challenge, meanwhile, a feature extraction module is designed to capture the local and global feature information.

The remainder of this paper is organized as follows. Section 2 summarizes related works about VFI and multi-scale strategy in the VFI field. Section 3 presents the proposed approach, including the MSEConv, the network architecture, and the training strategy. The experimental results are provided in Section 4, and concluding remarks are made in Section 5.

2 RELATED WORK

All kinds of methods to synthesize intermediate video frames have been introduced. In this section, we provide an overview of recent VFI methods and multi-scale strategies in the VFI field in the following.

2.1 Video Frame Interpolation (VFI)

VFI is a common yet challenging computer vision task. In recent years, VFI has been extensively studied and many works have occurred. According to the strategy of motion modeling, existing VFI methods can be roughly categorized into the traditional VFI algorithms and the convolution operation-based VFI approaches.

The traditional VFI works estimate the dense optical flow maps using optical flow algorithms [12, 16, 19, 43, 46], and then warp the input frames [4, 15, 47, 50], or regard video frames as linear combinations of wavelets with different directions and frequencies [33, 34]. Thereinto, a phase-based approach is proposed to represent motion in the phase shift of individual pixels [34]. This method combines phase information across the levels of a multi-scale pyramid. Whereafter, Meyer et al. proposed PhaseNet [33] to robustly handle challenging scenarios such as large and non-linear motions. The PhaseNet directly estimates the phase decomposition of the intermediate frame, thus, it has achieved prominent successes in image generation and synthesis. In addition, He et al. [15] designed a spatiotemporal saliency-based VFI approach. In this method, a spatiotemporal saliency model is first designed to select salient frames. Then, two motion vector refining schemes is adopted for high and low saliency frames to hierarchically refine the motion vectors, respectively. In the end, image enhancement is performed for salient frames to produce high-quality interpolated frames. Unfortunately, these methods are less effective in complex scenes due to their incapability of accurately estimating the optical flow or representing high-frequency components.

Recent works have demonstrated the success of applying deep neural networks (DNN) in the field of VFI [10], which uses the convolution operation to model an object's motion. They can further be classified into several categories: the optical flow-based methods, the Generative Adversary Networks (GAN)-based methods, and the kernel-based methods. The performance of optical flow-based methods is largely derived from the accuracy of the estimated optical flow. Great progress has been continuously made in optical flow estimation lately. With the advances in DNN, a FlowNet [19] is presented to learn the optical flow with DNN in a supervised learning way. Dutta et al. [13] designed a space-time convolution network to approximate the per-pixel motion. Specifically, an end-to-end 3D CNN encoder-decoder architecture over bidirectional optical flows and occlusion maps was developed to estimate the non-linear motion model of each pixel. A separate neural module, IFNet [18], is directly adopted to estimate the intermediate optical flow, supervised by a privileged distillation scheme. Later, a novel VFI approach [37] is presented to consider the local location and intensity of exceptional motion. The DNN can estimate motion information, i.e. optical flow, more accurately, but, it is still learned in a supervised manner causing it not sufficient enough to handle motion in occlusion. Consequently, the interpolated intermediate frames still arise blurring on the edge of the moving object. Besides, there are also several countermeasures based on implicit motion information to interpolate video frames. Typically, Shen et al. [40] proposed a unified optimization framework for joint frame interpolation and deblurring with spatial degradations. Specifically, a frame interpolation module with a pyramid structure was developed to cyclically synthesize high-quality intermediate frames. Since existing VFI methods usually produce the final interpolated results with average solutions that are not clear enough, Zhou et al. [54] proposed a novel texture consistency loss to relax the strict constraint of the pre-defined ground truth and a cross-scale pyramid alignment to make better use of multiscale information, making it possible to generate much clearer details. A novel trajectory-aware Transformer for VFI (TTVFI) is proposed [28]. In this method, the warped features with inconsistent motions and relevant regions in a motion trajectory from two original consecutive frames are respectively formulated as query tokens, keys and values. Due to ineffectively extracting issue of inter-frame motion and appearance information, a new unified module [53] is proposed by using inter-frame attention to explicitly extract motion and appearance information. In addition, this module can be seamlessly integrated into any CNN or Transformer-based VFI methods. Recently, a novel VFI framework is presented [51]. In this method, to infer accurate intermediate motion, a focalized confidence-ware trajectory estimation is first proposed by learning reliable optical flow candidates and suppressing the outliers. Then, a range-nullspace synthesis is solving by learning decoupled components in orthogonal subspaces. A joint non-linear motion regression strategy [29] is presented to model the complicated inter-frame motions to deal with real-world motion cases, such as variable acceleration, camera movement, and irregular movement. A fully differentiable many-to-many splatting framework [17] is proposed to interpolate intermediate frames efficiently, in which includes multiple bidirectional flows, many-to-many splatting scheme, and flexible spatial selective refinement component.

Since GAN has achieved prominent successes in image generation and synthesis, it has also been brought into the VFI field. The FINNiGAN [24] is firstly designed to synthesize the interpolated frame. By utilizing a combination of the MS-SSIM, ℓ_1, and GAN losses, its produced frames far exceed the qualitative appeal of those synthesized by closely related algorithms. Subsequently, an improved end-to-end GAN model [44] is proposed for VFI. In this method, the authors jointly considered an adversarial loss, reconstruction loss, and motion blur degradation to design a combined loss function. A multi-scale dense attention GAN [48] is presented by embedding the spatial attention module into the generator to model long-range dependency for dynamic motion regions and using the discriminator to maintain temporal and spatial consistency.

Instead of using optical flow and GAN, another major trend in this research is kernel-based methods [6, 9, 25, 35, 36, 52]. They treat the frame interpolation issue as a convolution operation by utilizing a predicted kernel to convolve through two input frames. As a pioneer work, Niklaus et al. [35] presented the first kernel-based algorithm, that combines motion estimation and pixel synthesis into a one-step process. Specifically, it first adopts a DNN to estimate spatially-adaptive convolution kernels, which capture both the local motion and the coefficients for pixel synthesis; then the pixel synthesis is considered as local convolution over two input frames. However, because this method predicts each kernel for each pixel of the input frame, it is computation-expensive and memory-intensive. A flow-free method, FLAVR [22], which utilizes 3D spatial-temporal kernels to learn motion properties from unlabeled videos, is proposed for multi-frame video interpolation with completely end-to-end training way. Subsequently, Cheng et al. [5, 6] introduced a deformable separable convolution to obtain the kernels, offsets, and masks for interpolation information with much fewer but more relevant pixels. But, these methods have a shared weight for all positions, and it is not suitable for video because there are various motions in each position of a frame [25]. A new warping module, AdaCoF [25], is designed for VFI. It jointly predicts both the local kernels and offset vectors for each pixel of the interpolated frame. Compared with previous approaches, AdaCoF is one of the most generalized warping frameworks by introducing an independent frame warping module, and a dual-frame adversarial loss. Ding et al. [8] also proposed a compression-driven network for frame interpolation (CDFI) with multiple stages of training, which leverages model pruning via sparsity-inducing optimization to significantly reduce the model size while achieving superior performance. Recently, a lightweight-driven video frame interpolation network [9, 52] was proposed. Concretely, the bidirectional encoding structure with channel attention cascade was first introduced to better characterize the motion information for better visual appearance; then the local network lightweight idea was adopted into the previous structure to significantly eliminate its redundant parts of the model parameters. However, the motion beyond maximum offset cannot be well processed. Although these methods can improve the performance of the interpolation frame by setting a larger kernel, there is data redundancy. In contrast, our proposed MSEConv keeps enough sampling region and offsets by using multiple smaller kernels with different expandable and adaptive weight allocations to reduce the data redundancy leading to more efficiency and effectiveness. The training process of our proposed MSEConv is also end-to-end. Furthermore, the kernel-based methods and the conventional optical flow-based methods are proved to be specific instances of the proposed MSEConv.

2.2 Multiple Scale Strategy

The multiple scale strategy is to obtain more information from different scales for desirable performance. In existing VFI methods, a multiple scale strategy is achieved in a coarse-to-fine manner in frame-level [26, 39, 48], or steerable pyramid filters [33]. A PhaseNet [33] is proposed to robustly handle challenging scenarios such as large and non-linear motions. It consists of a DNN decoder to directly estimates the phase decomposition with steerable pyramid filters for the intermediate frame. A multi-scale dense attention GAN [48] is also proposed for VFI. The multi-scale strategy was first utilized in the generator to capture global and local information in a coarse-to-fine manner. Then, an attention module was designed to focus on the moving objects. Finally, a discriminator is introduced to assist the generator to capture the spatiotemporal consistency in a video sequence. A lightweight VFI network, namely FI-Net [26], is proposed to output the intermediate frame of arbitrary size. It utilizes a multi-scale strategy on the frame level to compute optical flow at the feature level for handling large motions and long-range dependencies. Another multi-scale framework synthesis scheme based on the transformer [31, 41] is also to obtain information at different scales on the feature layer. A multi-scale strategy is also introduced into IM-Net [39] for the composition of the high-resolution interpolation frame. It uses an economically structured architecture and end-to-end training with pyramid-tailored losses. In particular, the interpolated motion estimation is formulated as classification rather than regression. Our inspiration comes from this way, but we estimate the multi-scale expansion of small kernels instead of the multi-scale processing or the feature extraction of input frames. Thus, this warping framework only needs a smaller model.

3 THE PROPOSED APPROACH

In this section, we introduce our proposed unified warping framework, MSEConv, for VFI, including the construction of the multi-scale expandable deformable convolution (MSEConv), the details of our network architecture, and our training strategy.

3.1 Problem Definition

Given two consecutive video frames I_1, and I_2, our goal is to interpolate a frame I_{out} between these two frames. For each $I_{out}(x, y)$ to be synthesized, some kernels are estimated to sample the surrounding pixels of the corresponding position (x, y) in the input frame I_1 and I_2 respectively. We define the sampling operation as S. The interpolation frame process can be formulated as:

$$I_{out} = S(I_1) + S(I_2) \tag{1}$$

The task of kernel-based VFI algorithms produces a problem of how to find the most efficient sampling operation S. In this work, we design a new warping framework for sampling called **M**ulti-**S**cale **E**xpandable deformable **Conv**olution (MSEConv), which convolves the input frame with adaptive multiple small kernel weights of different expanding and offset vector for each pixel.

3.2 Multi-Scale Expandable Deformable Convolution (MSEConv)

Let the frame sampled from I be $\hat{I}$. When S is defined as a classic convolution operation, $\hat{I}$ can be written as follows.

$$\hat{I}(i, j) = \sum_{\kappa=0}^{N-1} \sum_{\iota=0}^{N-1} W_{\kappa,\iota} * I(i + \kappa, j + \iota) \tag{2}$$

where N, $W_{\kappa,\iota}$, and (κ, ι) are the kernel size, the kernel weights, and the coordinates in the kernel, respectively. In AdaConv [35], the 2D kernels are employed, and the kernel size n is directly set as 41 to model the large motion. However, estimating so many numbers of kernels (41×41) requires an enormous computational burden. Subsequently, the 2D kernel is simplified with two approximate 1D kernel K_v, K_h, denoted as,

$$K(\kappa, \iota) = K_v(\kappa, \iota) * K_h(\kappa, \iota) \tag{3}$$

where $K(\kappa, \iota)$, K_h, and K_v are the 2D kernel, the horizontal and vertical 1D kernel, respectively.

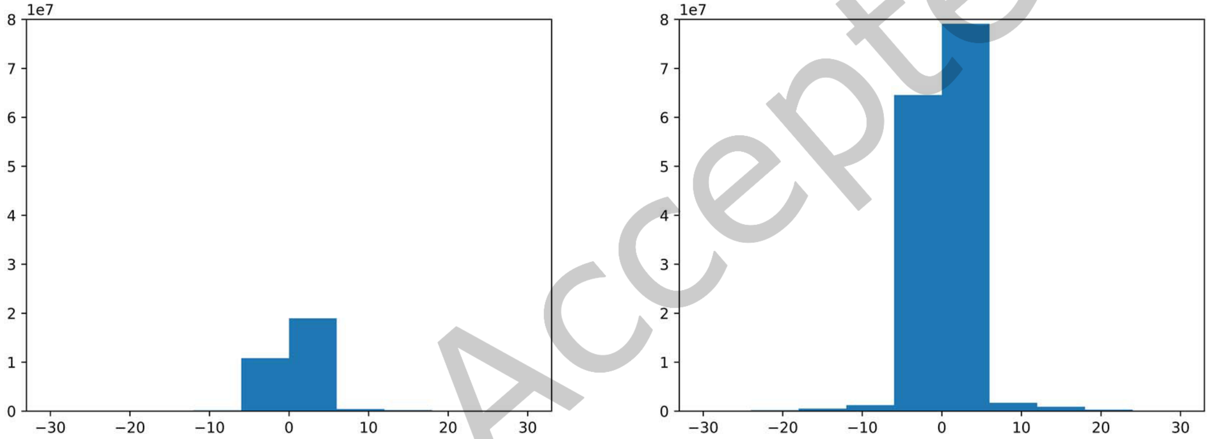
Although the number of kernels to be estimated is reduced from N^2 to $2N$, the number of the relevant pixel of sampling is still N^2. Later, the offset vectors $P_{\kappa,\iota} = (\alpha_{\kappa,\iota}, \beta_{\kappa,\iota})$ are added into the previous regular kernel to come up with the irregular kernel, defined as follows.

$$\hat{I}(i,j) = \sum_{\kappa=0}^{N-1} \sum_{\kappa=0}^{N-1} W_{\kappa,\iota} * I\left(i + \kappa + \alpha_{\kappa,\iota}, j + \kappa + \beta_{\kappa,\iota}\right) \tag{4}$$

Since this kind of irregular kernel has a shared weight for all positions, and each position of a frame has various motions, this irregular kernel is not suitable for video with large motion. Recently, we defined an improved irregular kernel [9, 52] with non-shared weights, which is borrowed from AdaCoF [25]. Here, the dilation coefficient d is integrated into the improved irregular kernel at the initial point of the offset vector, calculated as follows.

$$\hat{I}(i,j) = \sum_{\kappa=0}^{N-1} \sum_{\iota=0}^{N-1} W_{\kappa,\iota} * I\left(i + d\kappa + \alpha_{\kappa,\iota}, j + d\iota + \beta_{\kappa,\iota}\right) \tag{5}$$

where $d \in \{0, 1, 2, \cdots\}$.



(a) (b)

Fig. 2. The statistic results of offset vectors when the kernel size n is set as (a) 5 and (b) 11.

As reported in the existing irregular kernel-based VFI methods [5, 9, 25, 52], the kernel size of the irregular kernel is commonly set as 5 for greatly reducing the estimated numbers of the kernel and related pixels sampling, while the sampling area is randomly defined. We extend the kernel size to 11 for better modeling large motion in our previous works [9, 52]. However, in the real world video interpolation, we counted all the offset vectors $(\alpha_{\kappa,\iota}, \beta_{\kappa,\iota})$ and found that they had a certain range of offset values. The statistics result from the Middlebury dataset [1] are displayed in Figure 2. From this figure, we can easily observe that when the kernel size is 5, or 11, more than 98% of the offset vectors are in the internal $[-5, 5]$. Especially, when $n = 11$, the estimated offset vector in the range of $0 \sim 5$ is difficult to break through the range of the kernel size, while the estimated offset vector in the range of $6 \sim 11$ will produce large data redundancy inside. This also explains to some extent the phenomenon of the highest cost performance and why $N = 5$ is set as the optimal parameter in the existing irregular kernel-based VFI methods.

The proposed MSEConv is different from the sampling operation of the aforesaid methods, in which they simply expand the kernel size to obtain a larger sampling area. In the MSEConv, multiple small convolutions with different expansion degrees, which are described next, are used to sample the relevant pixels, and the number of actual sampling pixels is reduced from N^2 to $G \times n$, defined as follows.

$$\hat{\mathrm{I}}(\mathrm{i},\mathrm{j}) = \sum_{r=0}^{G-1}\sum_{\kappa=0}^{n-1}\sum_{\iota=0}^{n-1} W_{r,\kappa,\iota} * I\left(i + d_r\kappa + \alpha_{r,\kappa,\iota}, j + d_r\iota + \beta_{r,\kappa,\iota}\right) \tag{6}$$

where G, n, and d_r are respectively the group number of the kernel, the kernel size, and the dilation coefficient of the kernel group. $W_{r,\kappa,\iota}$ refers to the weight values of the different kernels, which are estimated for the unused pixel points. Meanwhile, $G, n \ll N$. The kernel size is decreased from N to n. Take the AdaCoF [9, 25, 52] as an example, the optimal kernel size is 11 for the large motion, and the number of actual sampling pixels is 121. Here, the G and n in the MSEConv are usually all set as 3, then the sampling pixels are 27. This further indicates that the MSEConv only needs fewer sampling pixels for modeling the large motion causing it more efficient in sampling.

Furthermore, we also design the expanding step, and adaptive weight allocation to further improve the ability of complex motion modeling of the MSEConv. The details are as follows.

Expanding Step To avoid data redundancy to the greatest extent, we introduce a certain expanding step $S \in \{1, 2, 3, 4, \dots\}$ into the estimated kernel for increasing the expansion of kernel, denoted as.

$$\hat{\mathrm{I}}(\mathrm{i},\mathrm{j}) = \sum_{r=0}^{G-1}\sum_{\kappa=0}^{n-1}\sum_{\iota=0}^{n-1} W_{r,\kappa,\iota} * I\left(i + (d_r + S)\kappa + \alpha_{r,\kappa,\iota}\ , j + (d_r + S)\iota + \beta_{r,\kappa,\iota}\right) \tag{7}$$

where the default value of d_r is 1.

Adaptive Weight Allocation In the real-world video, the proportion of moving regions is relatively small, which means that most pixels do not need large convolutional kernels to sample remote-related pixels. Therefore, we adaptively assign weights to the pixels of the interpolated frames, which are synthesized by kernel sampling with different expansion degrees, to better distinguish motion and background regions, expressed as follows.

$$\hat{\mathrm{I}}(\mathrm{i},\mathrm{j}) = \sum_{r=0}^{G-1}\sum_{\kappa=0}^{n-1}\sum_{\iota=0}^{n-1} W_{awa} * W_{r,\kappa,\iota} * I\left(i + (d_r + S)\kappa + \alpha_{r,\kappa,\iota}\ , j + (d_r + S)\iota + \beta_{r,\kappa,\iota}\right) \tag{8}$$

where W_{awa} is the adaptive weight, and $\sum_{r=0}^{G-1} W_{awa} = 1$.

The Reasoning of The Unified Framework In the MSEConv, both the kernel-based methods [5, 6, 9, 25, 35, 36, 52] and the optical flow-based methods are specific instances. Based on the construction of the MSEConv, we can observe that the MSEConv is derived from regular kernels to irregular kernels by gradually introducing the adaptively estimated kernels, offsets vectors, the dilation coefficient, the group number of the kernel, the expanding step, and the adaptive weight allocation. This process is also the gradual evolutionary of the literatures [5, 6, 9, 25, 35, 36, 52]. In Eq. 8, it is easy to make out that when $W_{ada} = 1$, $S = 0$, and $G = 1$, the interpolation process is the same as the one in [9, 25, 52]. Furthermore, when $W_{ada} = 1$, $S = 0$, $G = 1$, and $d_r = 0$, this process belongs to the definition of the irregular kernel in the literature [5, 6]. In the same way, when $W_{ada} = 1$, $S = 0$, $G = 1$, $d_r = 0$, $\alpha_{r,\kappa,\iota} = 0$, and $\beta_{r,\kappa,\iota} = 0$, this changes to the regular kernel, that is the core idea of the literature [35, 36].

On the other hand, specifically, we can use the 1D kernels to approximate 2D kernels, the process is expressed as

$$\hat{\mathrm{I}}(\mathrm{i},\mathrm{j}) = \sum_{r=0}^{G-1}\sum_{\kappa=0}^{n-1}\sum_{\iota=0}^{n-1} [W_{awa} * W_{v,r,\kappa,\iota} * W_{h,r,\kappa,\iota} * I\left(i + (d_r + S)\kappa + \alpha_{r,\kappa,\iota}\ , j + (d_r + S)\iota + \beta_{r,\kappa,\iota}\right)] \tag{9}$$

where $W_{v,r,\kappa,\iota}$ and $W_{h,r,\kappa,\iota}$ are the horizontal and vertical 1D kernel of the 2D kernel $W_{r,\kappa,\iota}$, respectively. when $G, n = 1$, $W_{ada} = 1$, $S = 0$, and $d_r = 0$, the kernel becomes a single point, and the patches are also single pixels.

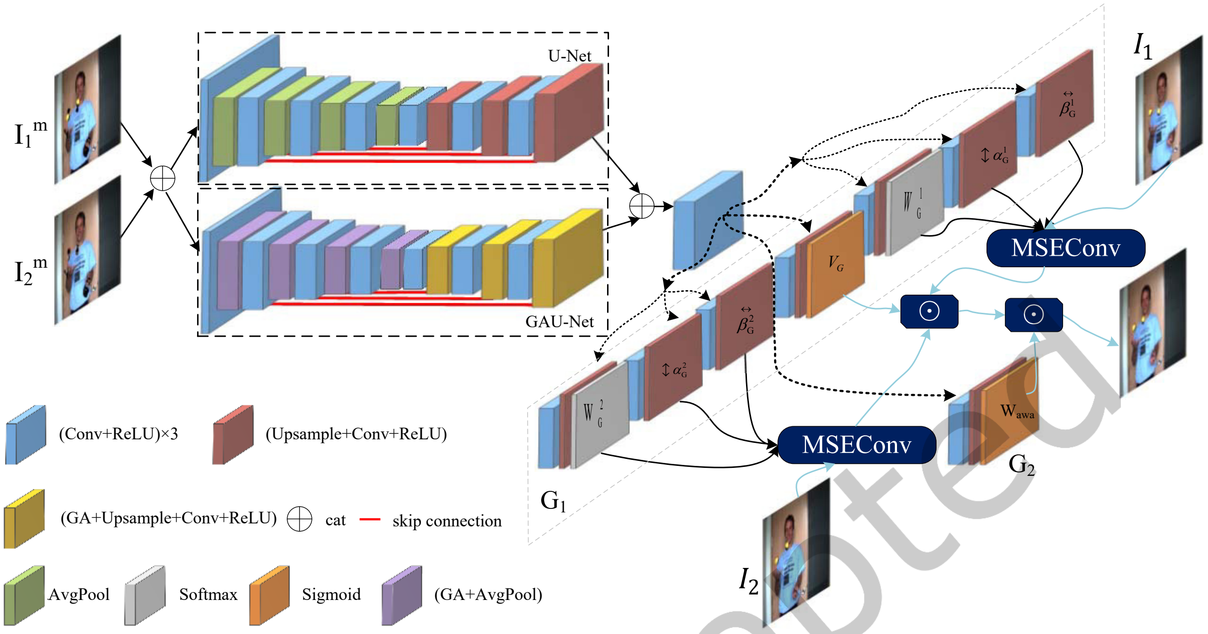


Fig. 3. The neural network architecture. The model consists of three main parts: the feature extraction module, sub-networks, and MSEConv. The feature extraction module includes the local feature extractor, U-Net, and the global feature extractor, GAU-Net. It characterizes input frames, which are given to sub-networks to estimate the parameters for MSEConv. The MSEConv synthesizes the intermediate frame using the input frames and parameters. Note that the frames I_1^m and I_2^m are the results after executing the operation of the mask occlusion, while the frames I_1, and I_2 are the original input frames.

The video frame interpolation process turns into the process of the bilinear interpolation of the single pixels [5]. Therefore, the vectors in Eq. 9 are changed to the scalars. In this case, it can be reformulated as

$$\hat{\mathrm{I}}(\mathrm{i},\mathrm{j}) = W_v * W_h * I(i+\alpha_{\kappa,\iota}, j+\beta_{\kappa,\iota}) \tag{10}$$

where $\alpha_{\kappa,\iota}$, and $\beta_{\kappa,\iota}$ denote learnable offsets. Each offset, and $W_v W_h$, which are the horizontal and vertical 1D kernel, can be considered as an offset component, and occlusion masks of optical flow, respectively. Obviously, Eq. 10 can be regarded as a bi-directional warping operation.

In a word, the proposed MSEConv can serve as the unified warping framework to represent existing kernel-based methods and conventional flow-based methods (Please refer to [5, 6, 9, 25, 35, 36, 52] for details).

3.3 Network Architecture

A full convolution neural network is proposed based on the designed unified warping framework. The whole network can be divided into the following three parts: the feature extraction module, the subnetwork, and the MSEConv as illustrated in Figure 3. The first part is designed to characterize input frames, the second part is used for parameter estimation, and the last part addresses the detailed design of the MSEConv in the previous subsection for synthesizing the intermediate frame by using the input frames and estimated parameters.

3.3.1 Feature Extraction Module. Like most kernel-based methods [5, 6, 9, 25, 52], we use the U-Net structure as the backbone. Due to the characteristics of local perception of DNN, the representation ability of global features is poor. Therefore, global attention (GA) is introduced into the U-Net structure from a global perspective, denoted as GAU-Net. The U-Net and GAU-Net are all composed of the standard convolution, average pooling, upsampling, and Relu activation functions. The standard convolution and Rule activation functions are general modules. The average pooling is only used for the encoder, while the upsampling is adopted for the decoder. In addition, a global attention operation is performed after each feature layer in the GAU-Net. The acquisition of the global feature is directly reconstructed the feature tensor $\mathfrak{R}1^{C\times W\times H}$ to $\mathfrak{R}2^{C\times WH}$ and the feature tensor $\mathfrak{R}1^{C\times W\times H}$ into $\mathfrak{R}3^{1\times W\times H}$ after a point convolution. Then, the reconstructed results perform softmax and matrix multiplication to produce the $\mathfrak{R}4^{C\times 1\times 1}$. Finally, the multiplied results are added to the initial feature tensor $\mathfrak{R}1^{C\times W\times H}$ after performing dimension reduction and dimension raising with the point convolution.

The proposed feature extractor module contains the local feature extractor, U-Net, and the global feature extractor, GAU-Net. The number of channels in the feature layer of the U-Net and GAU-Net is 32, 64, 128, 256, 256, 128, and 64, respectively. Finally, the feature vector with global information from the GAU-Net and the conventional feature vector from the U-Net are spliced together. The final feature vector is formed as the input of the sub-network through a basic convolution block. The detailed configurations of this subnetwork are provided in Table 1, in which $block\_0 \sim block\_6$ belong to the U-Net, while $block\_7 \sim block\_13$ are the GAU-Net.

Table 1. The detailed configurations of the feature extraction module, in which cat, AvgPool, UP, and GA mean the operations of the concatenation, average pooling, upsampling, and global attention, respectively.

Name	Input	The size of convolution	The In/Out of the channel	Rate
block_0	cat(frame0, frame1)	3×3	6/32	1
block_1	AvgPool(block_0)	3×3	32/64	1/2
block_2	AvgPool(block_1)	3×3	64/128	1/2
block_3	AvgPool(block_2)	3×3	128/256	1/2
block_4	AvgPool(block_3)	3×3	256/256	2
block_5	Up(block_4)+block_3	3×3	256/128	2
block_6	Up(block_5)+block_2	3×3	128/64	2
block_7	cat(frame0, frame1)	3×3	6/32	1
block_8	AvgPool(GA(block_7))	3×3	32/64	1/2
block_9	AvgPool(GA(block_8))	3×3	64/128	1/2
block_10	AvgPool(GA(block_9))	3×3	128/256	1/2
block_11	AvgPool(GA(block_10))	3×3	256/256	2
block_12	Up(GA(block_11))+block_10	3×3	256/128	2
block_13	Up(GA(block_12))+block_9	3×3	128/64	2
block_14	cat(Up(block_6), Up(GA(block_13)))+block_1+block_8	3×3	(64+64)/64	1

3.3.2 Subnetwork. The sub-network contains the G_1, and G_2 estimators. The former is used to estimate the kernel weights W, the occlusion graph V [25] and offset vectors $(\alpha_{\kappa,\iota}, \beta_{\kappa,\iota})$, while the latter is for the adaptive weight W_{awa}. W is nonnegative and $\sum W = 1$. V and W_{awa} use Sigmoid activation to satisfy $V \in [0,1]^{M\times N}$, and $\sum W_{awa} = 1$. More specific subnetwork structures are provided in Table 2.

Table 2. The detailed configurations of the subnetwork.

	Name	Input	The size of convolution	The In/Out of the channel	Rate
α/β	block_(α/β)	block_14	3×3	64/9	1
	output_(α/β)	Up(block_(α/β))	3×3	9/9	1
V/W_{awa}	block_(V/W_{awa})	block_14	3×3	64/9	1
	output_(V/W_{awa})	Sigmooid(Up(block_(V/W_{awa})))	3×3	9/9	1
W	block_W	block_14	3×3	64/9	1
	block_W	Softmax(Up(block_W))	3×3	9/9	1

Algorithm 1 Procedure of synthesizing

```
Input: Two consecutive video frames $I_1, I_2$
Output: The synthesizing frame $I_{out}$
function MSEConv($W, V, W_{awa}, \alpha_{\kappa,\iota}, \beta_{\kappa,\iota}$)
    for $i = 1, j = 1 \rightarrow i = N, j = N$ do
        Synthetic pixel $\hat{I}_1(i, j)$ based on Equation (8)
        Synthetic pixel $\hat{I}_2(i, j)$ based on Equation (8)
        $I_{out}(i, j) = \hat{I}_1(i, j) + \hat{I}_2(i, j)$
    end for
end function
```

3.3.3 MSEConv. The MSEConv utilizes the estimated kernel weights W, the occlusion graph V, the adaptive weight W_{awa} and offset vectors $(\alpha_{\kappa,\iota}, \beta_{\kappa,\iota})$ to adaptively convolve each input frame, synthesizing an intermediate interpolated frame. This process is adaptive and executed with Eq. 8 in this work, which does not totally resemble the operation presented in [5, 6, 9, 25, 52]. Finally, the synthesized frame is produced by adding the two previous interpolated frames. The algorithm of synthesizing procedure is described in Algorithm 1.

3.4 Training Strategy

Loss Function In this work, three loss functions have been combined to perform loss regression for the network so that the intermediate frame is more approximated to the real one. The ℓ_1 norm is employed as the first type of loss to reduce the difference between interpolation frames and real ones, expressed as.

$$\mathcal{L}_1 = \left\|\hat{I} - I_{gt}\right\|_1 \tag{11}$$

where I_{gt} means the ground truth of the real frame. Here, the ℓ_2 norm can also be used, but it is well known that the ℓ_2 norm will lead to blurring results in most image synthesis tasks [30, 32].

The second type of loss is perceptual loss, which is capable of effectively producing more realistic visual results [11]. The perceptual loss is usually based on the extracted advanced features, defined as:

$$\mathcal{L}_{per} = \left\|\phi(\hat{I}) - \phi\left(I_{gt}\right)\right\|_2^2 \tag{12}$$

where $\phi(\cdot)$ is the feature extractor. Note that the $\phi(\cdot)$ is usually adopted the *conv*4_3 of the VGG16 network pre-trained by ImageNet [21, 25] to extract advanced features from a frame.

The third loss is the adversarial loss [9, 25, 52]. It uses a discriminator D, which is trained on the interpolated frame and each input frame in the temporal order, to distinguish which of the two is the interpolated frame for the higher visual quality and sharpness result, calculated as.

$$\mathcal{L}_{adv} = D\left(\left[I_n, \hat{I_n}\right]\right) \log\left(D\left(\left[I_n, \hat{I_n}\right]\right)\right) + D\left(\left[\hat{I_{n+1}}, I_{n+1}\right]\right) \log\left(D\left(\left[\hat{I_{n+1}}, I_{n+1}\right]\right)\right) \tag{13}$$

The final loss function is summarized as.

$$\mathcal{L}_{all} = \lambda_1 \mathcal{L}_1 + \lambda_{per} \mathcal{L}_{per} + \lambda_{adv} \mathcal{L}_{adv} \tag{14}$$

Mask Occlusion Training Strategy In the task of VFI, there also exist motion occlusions in addition to large motions. Although the DNN-based VFI methods [7, 48] can implicitly handle the motion occlusion in the input frame to some extent, there is plenty of room for the improvement of motion occlusion. Therefore, we further introduce a mask occlusion training strategy to enhance the robustness of the network to the challenge of motion occlusion. We first randomly occluded the part of the motion regions of the input frames in the training dataset. Then, these occluded regions are masked and concatenated as the input of the proposed network. But the input frame of sampling is original. Thereinto, the regions of the motion occlusion are manually operated. As a consequence, this strategy has increased the robustness of the network against the motion occlusion. The masked motion occlusion is displayed in Figure 4. From this figure, we can observe that the motion occlusion is correctly marked. We can also further infer that after inputting these frames with masked motion occlusion into the network, they can assist the network in effectively modeling these motion occlusions, which is further verified in the experiments section.

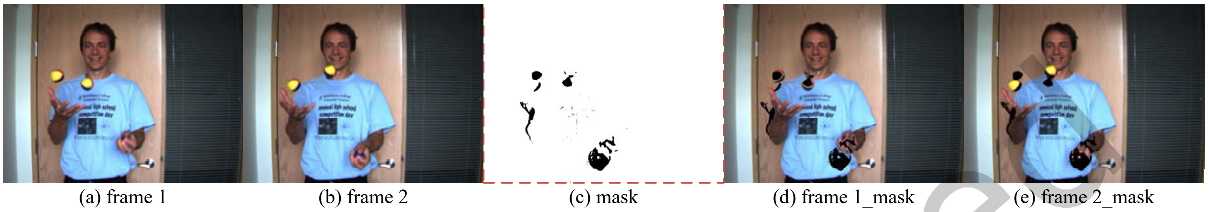

(a) frame 1 (b) frame 2 (c) mask (d) frame 1_mask (e) frame 2_mask

Fig. 4. The masked results of the motion occlusion. (a) and (b) are the original frames; (c) is the occlusion mask; (d) and (e) are the frames with masked motion occlusion.

4 EXPERIMENTS

In this section, we first introduce the datasets and evaluation metrics. Then, comprehensive ablation studies are conducted to analyze the contributions of some important components. Finally, we compare the proposed MSEConv with state-of-the-art methods, especially most of the kernel-based approaches.

4.1 Datasets and Evaluation Metrics

The test datasets used for the experiments are the Middlebury dataset [1], and some randomly sampled sequences from UCF101 [42]. The training dataset is Vimeo90K [49]. We evaluate each algorithm/model by measuring PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural SIMilarity index) [45] for all test datasets.

Vimeo-90K: Vimeo-90K [49] dataset is a high-quality video clip dataset that contains over 89000 videos of 720p or higher resolution, which are downloaded from the Vimeo video-sharing platform. Finally, the Vimeo-90K dataset for training consists of 51312 triplets, each of which has three consecutive video frames with a resolution of 448 × 256. It contains multiple scenes with high-quality motion.

Middlebury: Middlebury [1] dataset is a benchmark dataset for optical flow that is widely used to evaluate video frame interpolation techniques. There are two subsets of the dataset: the Evaluation set and the Other set. Since only the Other set provides the factual intermediate frame, we use the Other set as test benchmarks. The resolution in this dataset is 640 × 480.

UCF101: UCF101 [42] dataset is composed of real users uploading videos. It has different categories of 101 sequences of different actions, including camera motion and cluttered background. Most images in the UCF101 dataset are partially moving.

PSNR: It is a common metric used for calculating image quality. The higher the ratio, the better the quality of the interpolated frame.

SSIM: It is used for measuring image quality by computing the perceptual difference between the original frame and the interpolated frame. It is based on the visible structures of the frames, thus it can only judge how much they differ from each other instead of deciding which one is better.

In addition, we also evaluate the size of our model by the number of parameters, and also the inference time.

4.2 Training Details

Vimeo90K [49] is chosen as the training dataset. It contains 51312 video frame triples of 256 × 448, in which the first and third frames in each triplet are used as the inputs, and the second frame is used as the ground truth for the intermediate frame. We randomly crop the original image to 256 × 256 to train the network, which is the same for other methods (AdaCoF [25], SepConv [5, 6], L^2BEC^2 [9, 52], CAIN [7], and RRIN [27]), for fair comparison. We also enhance the data by horizontal flipping and vertical flipping. Meanwhile, the order of input frames is randomly exchanged with a probability of 0.5 to eliminate the prior bias, as done by L^2BEC^2 [9, 52], and AdaCoF [25]. In addition, we adopt the mask occlusion strategy to improve the robustness of motion occlusion.

The proposed MSEConv is implemented on the Pytorch platform [38] and trained on an NVIDIA Geforce RTX 2080TI GPU. The AdaMax optimizer [23] is employed to train our neural network to ensure its convergence, where $\beta 1$, $\beta 2$, the initial learning rate, and batch size are set as 0.9, 0.999, 0.001, and 4, respectively. It decays half every 20 epochs, and the entire network is trained for 100 epochs.

4.3 Ablation Study

To further analyze the contributions brought by each module introduced in the proposed MSEConv method, comprehensive ablations are performed under five aspects, namely the feature extraction module, the number of the group of the kernel, expanding step, weight allocation, and the mask ratio of motion occlusion.

Table 3. Experimental result of the model with different extracted feature information.

	dataset	Middlebury		UCF101	
		PSNR	SSIM	PSNR	SSIM
U-Net(basic)	Vimeo90K	36.181	0.961	35.006	0.966
GAU-Net	Vimeo90K	35.865	0.961	34.699	0.964
MSEConv	Vimeo90K	36.269	0.963	35.104	0.966

Feature Extraction Module The feature extraction module of the MSEConv contains the local and global feature information. To show the effectiveness of complementarity of these two kinds of feature information, we respectively construct the local feature extraction module, and the global feature extraction module, namely, the basic U-Net, which has the pure convolution, and the variant GAU-Net, which has a global attention mechanism. The table 3 reports the experimental results. Apparently, MSEConv outperforms two other models. The results also show that the global and the local feature information complement each other to improve the model performance.

Table 4. Experimental result on the group number of kernel G.

Group	The number of sampling pixels	Middlebury		UCF101	
		PSNR	SSIM	PSNR	SSIM
G=1	9	35.380	0.955	34.873	0.965
G=2	18	35.870	0.960	34.960	0.965
G=3	27	36.269	0.963	35.104	0.966

Kernel Group The proposed MSEConv uses multiple small kernels with different expansions to replace the operation of expanding kernel size. Its core idea is to design the number of the group of the kernel, and the expanding step into the irregular kernel. To verify the effectiveness and superiority of this idea, we train models with different kernel groups $G \in \{1, 2, 3\}$, where the kernel size and the expansion step are all fixed as 3. As shown in Table 4, the more the number of the group of the kernel generally leads to better performance and its PSNR saturates as G increases.

Table 5. Experimental result on the kernel size n and the group number of kernel G.

The kernel size and the group number	The number of sampling pixels	Middlebury		UCF101	
		PSNR	SSIM	PSNR	SSIM
n=3,G=1	9	35.256	0.953	35.007	0.965
n=5,G=1	25	35.715	0.958	35.063	0.966
n=11,G=1	121	36.094	0.961	35.024	0.966
n=3,G=3	27	36.269	0.963	35.104	0.966

Further, we conduct another experiment on the relation between the kernel size with the group number. Firstly, we fixed the group number with 1, and increase the kernel size. Then, we fixed the kernel size and increase the group number. As reported in Table 5, increasing the group number of the kernel is more effective than expanding the kernel size, resulting in better performance about 1dB gain on the Middlebury dataset. From the third and fourth rows of Table 5, for the largest kernel, the relevant pixels needed to sample for each pixel synthesis is 121 (11×11), while for the MSEConv, the sampling pixels is 27 ($3 \times 3 \times 3$). But, the MSEConv achieves better performance with fewer sampling pixels about 0.17dB and 0.1dB gain on the Middlebury and UCF101 datasets, respectively. That is the MSEConv is more efficient in the sampling pixels.

Table 6. Experimental result on the expansion step S.

The expansion step	The number of sampling pixels	Middlebury		UCF101	
		PSNR	SSIM	PSNR	SSIM
S=1	27	36.096	0.962	34.029	0.965
S=2	27	35.977	0.961	34.918	0.965
S=3	27	36.269	0.963	35.104	0.966
S=4	27	36.090	0.961	35.019	0.965

Expansion Step As illustrated in Figure 2 of Section 3.2, 98% of the value of the estimated offset vector is in the range of $[5, 5]$. To reduce data redundancy and avoid data sparsity, the expansion step should be in the range of $[2, 3]$ [9, 25, 52]. Therefore, we train the models with different expansion steps $S \in \{1, 2, 3, 4\}$, where the kernel size and the kernel groups are all set as 3. The quantitative results are displayed in Table 6. The experiment is in agreement with the prediction of the expansion step. That is, the model with $S = 3$ achieves the best PSNR and SSIM results on each dataset.

Table 7. Experimental result on the weight allocation W_{ada}.

Weight allocation	Middlebury		UCF101	
	PSNR	SSIM	PSNR	SSIM
$w/o\ W_{ada}$	35.591	0.959	34.941	0.965
$w\ W_{ada}$	36.269	0.963	35.104	0.966

Adaptive Weight Allocation Since the proportion of moving regions is relatively small, the motion and background regions should be allocated different weights, while the motion regions should be highlighted in the interpolated frames. Therefore, we train another network without weight allocation to further test the effectiveness of adaptive weight allocation for the estimated multiple convolution kernels. The results are demonstrated in Table 7, in which w/o, and w mean the model without weight allocation, and with weight allocation, respectively. Obviously, the model with weight allocation has a performance advantage. This is because the MSEConv adaptively distinctions between motion and background regions. As a result, it obtains more important sampling pixels and accordingly weakens the impact of negative sampling pixels.

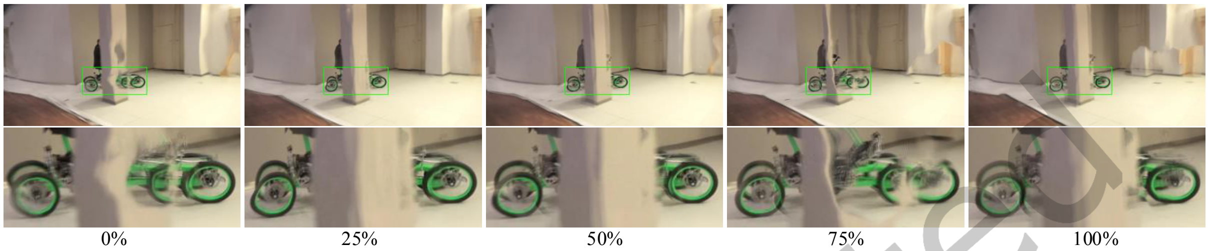


Fig. 5. Experimental result on the mask ratio of motion occlusion. From left to right, the mask ratios are 0%, 25%, 50%, 75%, and 100%, respectively.

The Mask Ratio of Motion Occlusion As is well known, there exactly exist motion occlusions in video sequences. In this work, we alleviate the negative effect of motion occlusion through the mask occlusion training strategy. Here, to verify the effectiveness of mask occlusion, an extra experiment is designed to analyze the mask ratio of motion occlusion. We occlude the motion regions of the input frame with different ratios (0%, 25%, 50%, 75%, 100%). An example is displayed in Figure 5, in which the first row is the interpolated frame by the MSEConv trained on the different occlusion ratio, and the second row is the larger version of the green box in the first row. From this figure, we can observe that the appropriate mask ratio of motion occlusion improves the ability of the network against the motion occlusion and further enhances the visual quality of interpolated results.

4.4 Comparison Results

We compare the MSEConv with the simple overlapping operation and the state-of-the-art VFI methods, involving SepConv [36], AdaCoF [25], L^2BEC^2 [9, 52], FLAVR [22], RRIN [27], RIFE [18], FME [51] and CAIN [7]. SepConv, AdaCoF, and L^2BEC^2 are the kernel-based approaches that address motions as the convolution operation. FLAVR uses 3D spatial-temporal kernels to learn motion properties. RRIN, RIFE and FME are optical flow-based models which calculate motions as pixel displacement. CAIN uses pixel shuffle and channel attention for pixel-level frame interpolation. Besides, the optimized version of the AdaCoF, namely AdaCoF+, is also employed for comparison. The experimental results are either produced by the open-source codes of the comparative VFI methods or directly cited from the reported references due to the lack of source code.

Quantitative Evaluation The quantitative evaluation is performed on the Middbury_other, and UCF101 dataset by measuring the PSNR and SSIM metrics of each VFI method. The results are shown in Table 8, in which the number of parameters for each model is computed by utilizing the in-line variable “model.parameters" or the in-line function “torchsummary" in pytorch, and the inference time is recorded as the average result for the inferring time of 100 randomly interpolated frames.

From Table 8, we observe that MSEConv achieve the best, and sub-optimal performance on Middbury_other and UCF1010 dataset, respectively, yet with fewer the number of sampling pixel. Compared with the baseline AdaCoF, and AdaCoF+, the proposed MSEConv obtains about 0.514dB, and and 0.175 dB gains in terms of PSNR on Middbury_other dataset, while has only 0.041 dB and 0.080 dB gains on UCF101 dataset. This is because the

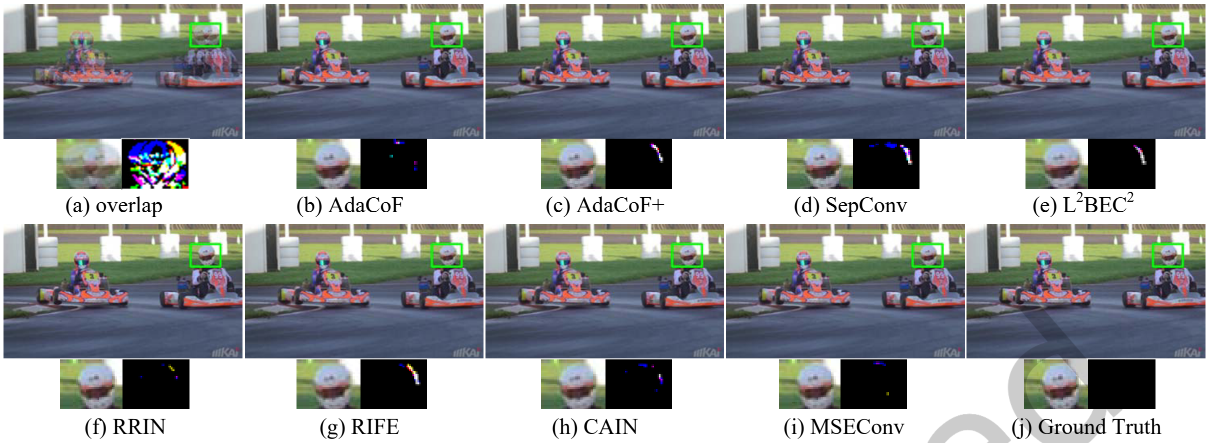
Table 8. Quantitative comparisons with state-of-the-art methods on the Middlebury, and UCF101.

	Training dataset	The number of sampling pixel	Middlebury		UCF101		Parameters (million)	Inference time(ms)
			PSNR	SSIM	PSNR	SSIM		
SepConv	pre-trained	2601 (51×51)	35.163	0.953	34.258	0.962	21.68	44.37
AdaCoF	Vimeo-90K	25 (5×5)	35.715	0.958	35.063	0.966	21.84	41.55
AdaCoF+	Vimeo-90K	121 (11×11)	36.094	0.961	35.024	0.966	22.93	120.40
L^2BEC^2	Vimeo-90K	25 (5×5)	35.841	0.961	35.120	0.966	7.05	44.67
RRIN	pre-trained	-	35.978	0.960	34.968	0.966	19.19	81.86
RIFE	pre-trained	-	34.804	0.951	34.568	0.963	3.03	21.79
CAIN	Vimeo-90K	-	35.100	0.950	34.964	0.965	42.78	489.32
FLAVR	Vimeo-90K	-	36.250	0.958	33.310	0.971	16.60	45.48
FME	Vimeo-90K	-	-	-	33.250	0.970	10.10	93.61
Ours	Vimeo-90K	27 ($3 \times 3 \times 3$)	36.269	0.963	35.104	0.966	7.40	96.11

UCF101 usually has small or medium motions, this case can also be modeled by the smaller kernel size. On the UCF101 dataset, the MSEConv is comparable to L^2BEC^2 for PSNR metric, with only an 0.016 performance drop. Furthermore, our proposed method is also comparable to FLAVR and FME with tiny drop for SSIM metric. This is because on the task of single frame interpolation, Middleburry has test samples with only two input frames while FLAVR requires 4 frame inputs. Thus, its results is suboptimal. And, the goal of FME is designed for the interpolation of 4K videos, thus, slightly worse performance is exhibited on low resolution datasets UCF101. The parameters of MSEConv are less than that of the comparable VFI methods, except for the RIFE and L^2BEC^2. But, the goal of the RIFE and L^2BEC^2 are to deploy them on small terminals by adopting the lightweight strategy. Thus, the core idea of these two methods is to reduce the parameters of the model. Although RIFE has fewer parameters than MSEConv, MSEConv is superior to RIFE in terms of the objective index on the Middlebury and UCF101 datasets. For the inference time, MSEConv consumes more time than the kernel-base methods due to the extra computation of the adaptive weight allocation, and kernel group. However, it is faster than the optimized version of the AdaCoF, namely AdaCoF+, and CAIN.

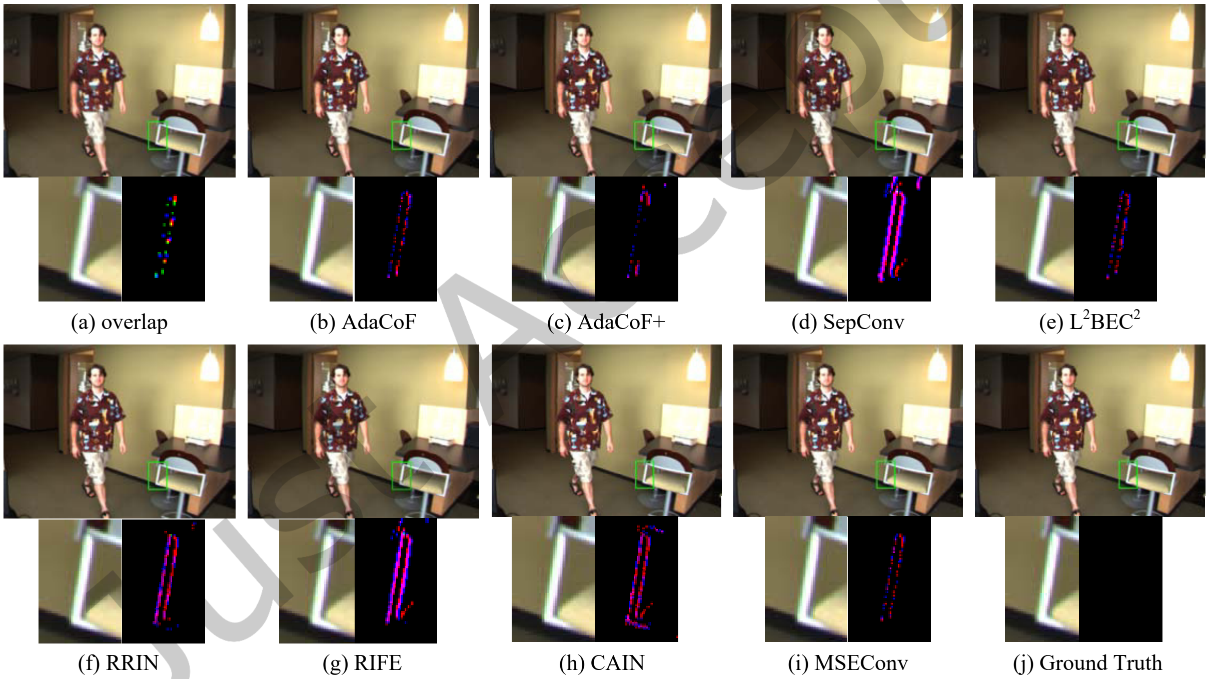
In short, compared with the kernel-based methods, MSEConv obtains a performance gain with only 22.3% of the number of sampling pixels to that of the AdaCoF+ and equivalently sampling pixel against the AdaCoF and L^2BEC^2 on the public datasets.

Visual Comparison Besides quantitative evaluation, we also compare the visual qualities of interpolated frames. Fig. 6, and Fig. 7 report the interpolated frames by different VFI methods on the Middbury_other, and UCF101 dataset. Two sequences “kart-turn", and “walking" have complex motions and repetitive patterns that easily lead to holes and blurring, they are commonly chosen as visually comparable samples. For better comparison, the driver of the kart, and the edge of the table marked by green boxes are enlarged, and the difference between the selected regions and the corresponding regions of the ground truth is also demonstrated on the right of the enlarged version. For the driver, the MSEConv has an obvious advantage when processing complex motions compared with kernel-based methods and other methods. Besides, the proposed method has the least difference values due to the desirable motion modeling benefiting from the construction of the MSEConv. For the edge of the table, the MSEConv retains more edge and contour information, whereas the other methods all produce more differences. In a word, we can conclude from the visual comparisons that MSEConv acquires better performance when dealing with complex motion and repetitive patterns. Moreover, MSEConv possesses a stronger ability to model motion than kernel-based methods. Since MSEConv obtains better-interpolated results with fewer sampling pixels, the effectiveness of MSEConv is verified.


(a) overlap (b) AdaCoF (c) AdaCoF+ (d) SepConv (e) L^2BEC^2

(f) RRIN (g) RIFE (h) CAIN (i) MSEConv (j) Ground Truth

Fig. 6. Visual comparison of the “kart-turn" sequence.


(a) overlap (b) AdaCoF (c) AdaCoF+ (d) SepConv (e) L^2BEC^2

(f) RRIN (g) RIFE (h) CAIN (i) MSEConv (j) Ground Truth

Fig. 7. Visual comparison of the “walking" sequence.

Here, we further check how the MSEConv and other state-of-the-art methods address the two main challenges in real videos: large motion and occlusion.

Large Motion When the reference point appears further away, the search regions should be expanded accordingly [9, 25, 52]. As a consequence, the problem of the large motion become one of the most challenging in the VFI research field. Figure 8 shows the interpolated results of the regions of large motion by various algorithms

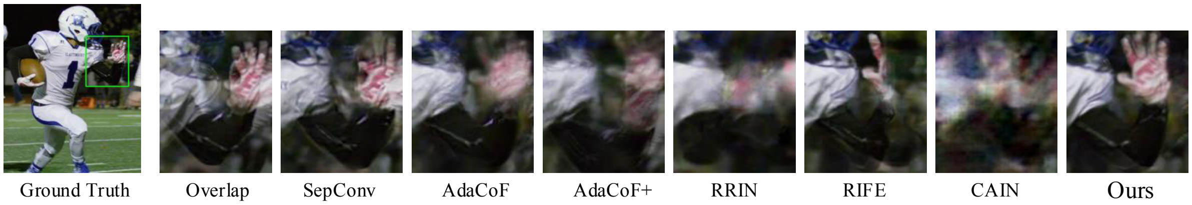


Fig. 8. Visual comparison of the “tackle" sequence with large motion.

including the MSEConv. The results of AdaCoF, RRIN, and CAIN tend to be blurry and SepConv, AdaCoF+, and RIFE have some hand contour, yet still suffer from some artifacts. Compared to the competing approaches, our method MSEConv better models the large motion, namely fast moving hand. In addition, our proposed method also relieves the motion blurs of the moving object.

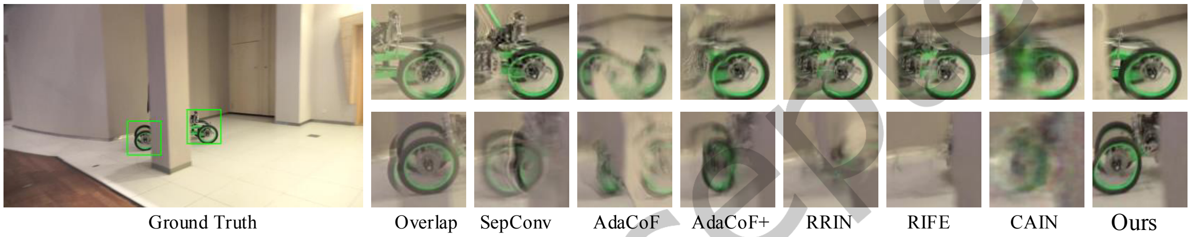


Fig. 9. Visual comparison of the “e-bike" sequence with occlusion.

Occlusion As we know, the video sequence exists in the temporal dimension, thus, the objects in one frame may arise in both adjacent frames. However, in case of occlusion, the objects do not appear in one of the frames [9, 25, 52]. The problem of occlusion is also challenging in the VFI research field. Figure 9 illustrates the interpolated results of the occluded bike by various methods. From the first row, we can observe that AdaCoF and CAIN cannot well reconstruct the front wheel of the bike. In the second row, due to the incomplete bike causing inaccurate optical flow estimation, the RRIN and RIFE cannot produce the correct interpolated results, while SepConv, AdaCoF, AdaCoF+, and CAIN still have varying degrees of blurring. In contrast, our method well handles the occlusion problem because of the integration of the mask occlusion training strategy.

5 LIMITATIONS AND FUTURE WORK

Although our proposed MSEConv has achieved a noticeable improvement, some limitations are still need to be investigated. First, the input frames of MSEConv is two adjacent frames without considering to leverage the long-range motion information from multiple consecutive frames. Second, the existing kernel-based methods still have some limitations in sampling, and the number and range of sampling required for moving regions and non-moving regions in video frames are inconsistent. Although the proposed method achieves different sampling ranges for different regions by adaptively assigning weight values to kernels with different degrees of expansion, the actual estimated number of samples is still the same. How to adaptively adjust the number of samples for different regions is still a problem to be solved based on the kernel method. In future work, we will take this perspective as the starting point, and further consider the time factor to achieve adaptive allocation sampling in space and time. Besides, we will consider to extend our proposed method with multiple frame inputs to fine-grained capture motion information for better results of video interpolation. Finally, we will also employ the texture consistency loss [54] in the spatial-temporal domain to overcome the dependence on the ground-truth frame and alleviate the blurring effect of averaged solutions.

6 CONCLUSION

In this paper, we propose a multi-scale expansion deformable convolution (MSEConv) for video frame interpolation. The key of this method is that in addition to using multiple small kernels for sampling, our method also allocates the expansion of different scales for each kernel and adaptively assigns weights to the pixels of the interpolated frames for more effectively conducting large-scale pixel sampling. This makes our method sample fewer pixels to process more complex motions. Meanwhile, we further extract features both locally and globally. In addition, our mask occlusion training strategy makes our network more robust when dealing with motion occlusion. As demonstrated, both the kernel-based and optical flow-based methods are special instances of the MSEConv and we perform favorably against the state-of-the-art on public datasets qualitatively and quantitatively with much fewer but more relevant sampling pixels.

7 ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (62272160, 62172059, U22A2030) and the Postgraduate Scientific Research Innovation Project of Hunan Province (CX20221058, CX20210811).

REFERENCES

[1] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. 2011. A database and evaluation methodology for optical flow. *International journal of computer vision* 92, 1 (2011), 1–31.

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

[3] Wenbo Bao, Xiaoyun Zhang, Li Chen, Lianghui Ding, and Zhiyong Gao. 2018. High-order model and dynamic filtering for frame rate up-conversion. *IEEE Transactions on Image Processing* 27, 8 (2018), 3813–3826.

[4] John L Barron, David J Fleet, and Steven S Beauchemin. 1994. Performance of optical flow techniques. *International journal of computer vision* 12, 1 (1994), 43–77.

[5] Xianhang Cheng and Zhenzhong Chen. 2020. Video frame interpolation via deformable separable convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10607–10614.

[6] Xianhang Cheng and Zhenzhong Chen. 2021. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[7] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. 2020. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10663–10671.

[8] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. 2021. CDFI: Compression-driven network design for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8001–8011.

[9] Xiangling Ding, Pu Huang, Dengyong Zhang, and Xianfeng Zhao. 2022. Video frame interpolation via local lightweight bidirectional encoding with channel attention cascade. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1915–1919.

[10] Jiong Dong, Kaoru Ota, and Mianxiong Dong. 2022. Video Frame Interpolation: A Comprehensive Survey. *ACM Transactions on Multimedia Computing, Communications and Applications* (2022).

[11] Alexey Dosovitskiy and Thomas Brox. 2016. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems* 29 (2016).

[12] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 2758–2766.

[13] Saikat Dutta, Arulkumar Subramaniam, and Anurag Mittal. 2022. Non-linear motion estimation for video frame interpolation using space-time convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1726–1731.

[14] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5515–5524.

[15] Jiale He, Gaobo Yang, Xin Liu, and Xiangling Ding. 2020. Spatio-temporal saliency-based motion vector refinement for frame rate up-conversion. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 2 (2020), 1–18.

[16] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. 2022. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3553–3562.

[17] Ping Hu, Simon Niklaus, Lu Zhang, Stan Sclaroff, and Kate Saenko. 2023. Video Frame Interpolation With Many-to-Many Splatting and Spatial Selective Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[18] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. 2020. RIFE: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294* (2020).

[19] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2462–2470.

[20] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 9000–9008.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision.* Springer, 694–711.

[22] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. 2023. Flavr: Flow-agnostic video representations for fast frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2071–2082.

[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[24] Mark Koren, Kunal Menda, and Apoorva Sharma. 2017. Frame interpolation using generative adversarial networks.

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

[26] Haopeng Li, Yuan Yuan, and Qi Wang. 2019. Fi-net: A lightweight video frame interpolation network using feature-level flow. *IEEE Access* 7 (2019), 118287–118296.

[27] Haopeng Li, Yuan Yuan, and Qi Wang. 2020. Video frame interpolation via residue refinement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2613–2617.

[28] Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian. 2023. Ttvfi: Learning trajectory-aware transformer for video frame interpolation. *IEEE Transactions on Image Processing* (2023).

[29] Meiqin Liu, Chenming Xu, Chao Yao, Chunyu Lin, and Yao Zhao. 2023. Jnmr: Joint non-linear motion regression for video frame interpolation. *IEEE Transactions on Image Processing* (2023).

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

[31] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. 2022. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3532–3542.

[32] Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440* (2015).

[33] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. 2018. PhaseNet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

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

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

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

[37] Minho Park, Hak Gu Kim, Sangmin Lee, and Yong Man Ro. 2020. Robust video frame interpolation with exceptional motion map. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 2 (2020), 754–764.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[39] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. 2019. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2398–2407.

[40] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. 2020. Video frame interpolation and enhancement via pyramid recurrent framework. *IEEE Transactions on Image Processing* 30 (2020), 277–292.

[41] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. 2022. Video frame interpolation transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 17482–17491.

[42] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

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

[44] Quang Nhat Tran and Shih-Hsuan Yang. 2020. Efficient video frame interpolation using generative adversarial networks. *Applied Sciences* 10, 18 (2020), 6245.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[46] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. 2013. DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*. 1385–1392.

[47] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. 2011. Optical flow guided TV-L 1 video interpolation and restoration. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 273–286.

[48] Jian Xiao and Xiaojun Bi. 2020. Multi-scale attention generative adversarial networks for video frame interpolation. *IEEE Access* 8 (2020), 94842–94851.

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

[50] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen Chen. 2013. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 7 (2013), 1235–1248.

[51] Zhiyang Yu, Yu Zhang, Dongqing Zou, Xijun Chen, Jimmy S Ren, and Shunqing Ren. 2023. Range-Nullspace Video Frame Interpolation With Focalized Motion Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22159–22168.

[52] Dengyong Zhang, Pu Huang, Xiangling Ding, Feng Li, Wenjie Zhu, Yun Song, and Gaobo Yang. 2023. L^2BEC^2: Local lightweight bidirectional encoding and channel attention cascade for video frame interpolation. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 2 (2023), 1–19.

[53] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. 2023. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5682–5692.

[54] Kun Zhou, Wenbo Li, Xiaoguang Han, and Jiangbo Lu. 2023. Exploring motion ambiguity and alignment for high-quality video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22169–22179.

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