Abstract—Flow-based and deformable convolution (DConv)-based methods are two mainstream approaches for solving the video frame interpolation (VFI) problem, which have made remarkable progress with the development of deep convolutional networks over the past years. However, flow-based VFI methods often suffer from the inaccuracy of flow map estimation, especially in dealing with complex and irregular real-world motions. DConv-based VFI methods have advantages in handling complex motions, while the increased degree of freedom makes the training of the DConv model difficult. To address these problems, in this article, we propose a flow guidance deformable compensation network (FGDCN) for the VFI task. FGDCN decomposes the frame sampling process into two steps: a flow step and a deformation step. Specifically, the flow step utilizes a coarse-to-fine flow estimation network to directly estimate the intermediate flows and synthesizes an anchor frame simultaneously. To ensure the accuracy of the estimated flow, a distillation loss and a task-oriented loss are jointly employed in this step. Under the guidance of the flow priors learned in step one, the deformation step designs a new pyramid deformable compensation network to compensate for the missing details of the flow step. In addition, a pyramid loss is proposed to supervise the model in both the image and frequency domains. Experimental results show that the proposed algorithm achieves excellent performance on various datasets with fewer parameters.

Index Terms—Video frame interpolation, motion estimation, motion compensation, deformable convolution, distillation learning.

I. INTRODUCTION

VIDEO frame interpolation (VFI) is a classical low-level vision task to increase the frame rate of a video sequence [1]. Recent years have witnessed the rapid development of VFI empowered by the success of deep neural networks [2], [3], [4]. However, there remain some unsolved problems due to challenges like occlusions, large motion, and lighting changes. According to whether or not the optical flow information is incorporated, the existing VFI approaches can be roughly divided into two categories: 1) motion-based methods and 2) motion-free methods.

In recent years, significant progress has been made by motion-based methods [3], [5], [6], [7]. These methods first estimate the bidirectional optical flows and then synthesize the target frame by warping two successive frames forward or backward. In general, these motion-based methods can explicitly represent the motion and reach high fidelity in most cases. However, due to the basic assumptions of the optical flow estimation, e.g. smoothness and consistency, motion-based methods are inherently difficult to handle the interpolation of complicated dynamic scenes, including the regions suffering from occlusion, blur, or abrupt brightness change [8].

Motion-free methods circumvent the need for flow map estimation and consequently are not susceptible to the associated issues. The most representative motion-free methods are the kernel-based methods [9], [10], which directly generate the target intermediate frame by applying spatially-adaptive convolution kernels to the given frames. Kernel-based methods are effective, however, the rigidity of the kernel shape severely limits the types of motions that such methods can handle. Indeed, one may need to choose a very large kernel size to ensure enough coverage, which is highly inefficient.

Most recently, deformable convolution (DConv) [11] has been investigated in the VFI task to warp features and frames. Gui et al. [12] propose the Featureflow, which regards DConv as the universal version of motion flows. Different from the one-to-one flow warping, the DConv layer produces multiple offsets for each pixel, using a weighted average of them to predict the target pixel. Thus DConv’s offsets warping can be considered as many-to-one flow warping. Theoretically, DConv methods can collect more diverse information and produce more complex transformations, which should be more robust than single optical flow methods when handling complex motions. However, in practice, the increased degree of freedom makes the training of the DConv model more difficult, which limits their performance [13], [14].

In this article, we propose a flow guidance deformable compensation network (FGDCN) to overcome the above problems. In Fig. 1, we compare the sampling strategies of different VFI methods. Different from the existing methods, our method decomposes the sampling process into two steps: a flow step and...
a deformation step. The flow step estimates the intermediate flows to establish the motion correlations between the input frames and the target frame. Since the estimated flow may be inaccurate in some areas, the deformation step further employs the DConv layer to fully explore the ambient local information for compensating the missing details of the single-point flow sampler. The decomposed two-stage learning strategy establishes the connections between the motion-based methods and the DConv-based methods and combines the advantages of both. To be specific, the flow prior makes the DConv layer easier to train, while the DConv layer enhances the transformation ability of the motion-based method and helps it to handle complex motions and recover more detailed information.

The network structure of the proposed FGDCN also contains a flow step and a deformation step. The flow step first estimates the intermediate flows and the occlusion mask in a coarse-to-fine manner and then uses them to synthesize an anchor frame. Our coarse flow estimation network is inspired by [15], which contains a series of intermediate flow estimation blocks to estimate the intermediate flow directly. Subsequently, a pyramid flow refinement network is employed to gradually refine the coarse intermediate flows. The coarse flows are supervised by the distillation loss and the refined flows are supervised in a task-oriented manner. Using the learned accurate flows as guidance, in the deformation step, we first use a pyramid deformable network to compensate for the missing details of the flow step. Then, a new pyramid deformable compensation network (PDCN) is proposed to compensate for the missing details of the flow step. In the deformation step, a new pyramid deformable compensation network (PDCN) is proposed to compensate for the missing details of the flow step.

Pyramid reconstruction losses are introduced to supervise the network in both the image and frequency domains. Experimental results show that FGDCN has advantages in handling complex and fast motion videos and it outperforms most of the latest state-of-the-art methods with fewer parameters.

II. Related Works

A. Motion-Based Video Frame Interpolation

Video frame interpolation, aiming to synthesize intermediate frames between existing ones of a video, is a longstanding problem. The recent mainstream VFI methods are motion-based methods, which first estimate the bidirectional optical flows and then use them to warp the input frames. Liu et al. [2] introduced a deep network that produced 3D optical flow vectors and warped input frames by trilinear sampling. Recent works have explored a few strategies for improving the performance of such methods. These strategies include utilizing additional contextual information to interpolate high-quality results [16], developing unsupervised techniques by cycle consistency [17], detecting the occlusion by exploring the depth information [18], forward warping input frames using softmax splatting [5], leveraging the transformer [19], [20] to model the long-term dependencies, exploring more diverse information using many-to-many flow [21], [22], and constructing efficient architectures for speeding up the interpolation [7]. Although significant progress has been made by these motion-based VFI approaches, their basic assumption that the motions between neighboring frames are uniform linearity, seriously limits their performance in complex scenarios. Recently, a more accurate quadratic model was proposed in [4] to replace the naive linear model for estimating motion. Nevertheless, the complexities and irregularities of real-world motions cannot be completely captured by a simple mathematical model [8]. In this article, we employ a powerful pyramid deformable network to compensate for the defects of motion-based approaches.

B. Intermediate Flow Estimation

The intermediate optical flows are highly uncertain, thus how to estimate accurate intermediate optical flows from the consecutive frames is a crucial problem for motion-based VFI methods. When obtaining the intermediate flow, the target image can be obtained by simple backward warping. Jiang et al. [3] linearly combined optical flows between the input frames to approximate...
the intermediate flows. Bao et al. [23] proposed a flow projection layer to reverse the optical flows between the input frames and the target frame. Xue et al. [24] proposed a task-oriented flow estimation network to predict the intermediate flow directly. Recent VFI methods [7], [15] proposed to use the distillation strategy to promote the accuracy of the intermediate flow estimation and lead to large performance improvement. Inspired by these works, our method jointly uses the distillation strategy and the task-oriented flow estimation strategy to predict the intermediate flow in a coarse-to-fine manner.

C. Deformable Convolution

Dai et al. [11] proposed the deformable convolution network, where additional offsets were calculated to obtain information away from its regular local kernel neighborhood. Zhu et al. [25] further introduced a modulation mask to strengthen the capability of manipulating spatial support regions. Let \( p_k \) be the \( k \)-th sampling offset in a standard convolution with kernel size \( n \times n \). For example, when \( n = 3 \), we have \( p_k \in \{(-1, -1), (-1, 0), \ldots, (1, 1)\} \). We denote the \( k \)-th additional learned offset and mask at location \( p + p_k \) by \( \Delta p_k \) and \( \Delta m_k \) respectively. A deformable convolution can be formulated as

\[
F_{\text{out}}(p) = \sum_{k=1}^{n^2} w(p_k) \cdot F_{\text{in}}(p + p_k + \Delta p_k) \cdot \Delta m_k(p), \tag{1}
\]

where \( F_{\text{in}} \) and \( F_{\text{out}} \) represent the input and output features respectively, \( w \) denotes the kernel weights. Due to its powerful spatial adaption ability, deformable convolution has been widely used in various tasks such as video object detection [26], action recognition [27], semantic segmentation [28]. In low-level vision tasks, Tian et al. [29] firstly used the deformable convolution to align adjacent frames for solving the video super-resolution problem. Subsequently, Wang et al. [30] proposed a more powerful Pyramid, Cascading and Deformable (PCD) module for video frame alignment and achieved better results. Xiang et al. [31] proposed frame temporal interpolation network using the PCD module, which addressed the temporal interpolation and spatial super-resolution simultaneously. For VFI, [32] and [33] first introduced the DConv layer in kernel-based methods, where additional offsets were calculated to adaptively sample useful information from the input frames. Gui et al. and Chen et al. [12], [13] used the DConv layer to directly generate the intermediate frame by blending deep features. Shi et al. [8] proposed a more generalized DConv layer, which extracted the information from the multiple frames to solve the performance-limiting issues of conventional methods. Different from the above methods that directly use the DConv layer to model the complete motions, we employ a flow guidance DConv layer to compensate for the missing details of the motion-based methods.

D. Flow-Guided DConv Learning

Some works have proposed to use flow-guided DConv learning to improve the performance of the video super-resolution (VSR) and video inpainting (VI) tasks. Chan et al. [14] proposed an offset-fidelity loss to guide the offset learning with optical flow. This loss successfully avoids the overflow of offsets and alleviates the instability problem of deformable alignment in the VSR task. Subsequently, Lin et al. [34] and Chan et al. [35] proposed a flow-guided alignment layer to align the features from the reference frames to the target frames for the VSR task. Li et al. [36] designed a flow-guided feature propagation layer for the video inpainting (VI) task. Luo et al. [37] employed the flow-guided deformable alignment for improving the performance of burst super-resolution.

The above methods have demonstrated the benefits of the flow-guided DConv learning on VSR and VI tasks, however, it has not been exploited in the VFI task. As aforementioned, flow-based and the DConv-based VFI methods have their own strengths and weaknesses. A single model cannot cope with complex motions in real-world scenes. Therefore, how to properly unify the flow-based with the DConv-based approaches into one framework and take advantage of both will be very meaningful to solving the problems that VFI faces.

It’s worth noting that models [34], [35], [36], [37] designed for VSR and VI tasks focus on how to align features from the reference frame to the given target frame, while the VFI task focuses on how to synthesize the unknown target frame from its reference frames. The unknown target frame makes the estimated intermediate flows highly uncertain, as well as affects the subsequent DConv learning. To solve this problem, in this article, we propose a new VFI framework that contains two steps, i.e., flow step and the deformation step. The flow step employs a coarse-to-fine flow estimation network to estimate the intermediate flows directly and synthesize an anchor frame. The deformation step designs a new PDCN to compensate for the missing details of the flow step. Built upon deliberate thinking of the VFI task and well-designed network structures, we successfully unify two mainstream VFI methods (i.e., flow-based and DConv based) into one model and take advantages of both. Detailed network structure will be introduced in the next section.

III. PROPOSED METHOD

A. Network Overview

Given two input frames \( I_0 \) and \( I_1 \), the video frame interpolation is to synthesize an intermediate frame \( I_t \). To achieve this goal, we propose FGDCN for accurate frame interpolation. Its overall structure is shown in Fig. 2. It contains a flow step and a deformation step. The flow step first uses a coarse intermediate flow estimation network to predict the coarse intermediate flows \( \vec{v}_{t=0} \) and \( \vec{v}_{t=1} \). Then a pyramid network is employed to refine the coarse flows gradually and we get refined flows \( \vec{v}_{t=0} \) and \( \vec{v}_{t=1} \). The pyramid network contains three levels \( l \in \{1, 2, 3\} \). It extracts the pyramid features \( F^0_l \) and \( F^1_l \) in level \( l \) for frame \( I_0 \) and \( I_1 \), respectively. At the end of the flow step, we synthesize an anchor frame \( \vec{I}_t \) using the learned flows and occlusion mask. The deformation step contains a pyramid deformable compensation network (PDCN) and a multi-level frame synthesis network (MFSN). PDCN employs the DConv layers to compensate for the missing details of the flow step. MFSN uses the warped
Fig. 2. Overview of the proposed FGDCN. It contains a flow step and a deformation step. The flow step consists of a coarse intermediate flow estimation module, a pyramid flow refinement module, and an anchor frame interpolation module. The deformation step consists of a pyramid deformable compensation network and a multi-level frame synthesis network.

Fig. 3. (a) The structure of the coarse-to-fine flow estimation network. (b) The structure of the l-th flow refinement block (FRB).

pyramid features as input to learn the residuals between the anchor frame and the ground truth. Next, we will introduce each component in detail.

B. Flow Step

The flow step aims to estimate accurate flow maps and synthesize an anchor frame simultaneously. It contains three parts: a coarse intermediate flow estimation network (FEN), a pyramid flow refinement network (FRN), and an anchor frame interpolation operation. The coarse-to-fine flow estimation network is shown in Fig. 3(a). It first uses an FEN to estimate the coarse flows and then employs an FRN to refine them. In addition, we also utilize two loss functions, i.e., weakly supervised flow estimation loss and task-oriented flow estimation loss in this step to estimate the intermediate flow from coarse to fine.

1) Coarse Intermediate Flow Estimation: We employ FEN to predict the coarse intermediate flow and occlusion mask directly. In FEN, a series of intermediate flow estimation blocks (IFBs) are used to predict the coarse intermediate flows directly. The network is learned by a knowledge distillation scheme. In this article, we employ the pre-trained flow estimation network Liteflow [38] as the teacher model to produce the pseudo flow labels. Since the pseudo labels are not entirely accurate, we only use them for weak supervision in a downsampled resolution. Stronger supervision will be given in a task-oriented manner after the subsequent flow refinement network.

2) Pyramid Flow Refinement: Inspired by [6], [19], we employ a pyramid FRN to refine the coarse flows and the occlusion mask. The structure of the l-th flow refinement block is shown in Fig. 3(b). Firstly, we use the upsampled previous flows to warp the pyramid features and we get the l-th warped features $F_{l,0}^t$ and $F_{l,1}^t$. Then the warped features are fed into convolutional layers to update the occlusion mask $M^l$, which is also used to produce a temporary intermediate feature $F_{l}^t$. These operations can be represented as

$$M^l = M^{l-1} + \sigma_\tau(occ([F_{l,0}^t, F_{l,1}^t]))$$  \hspace{1cm}(2)

$$F_{l}^t = M^l \odot F_{0,0}^{l-1} + (1 - M^l) \odot F_{1,0}^{l-1}$$  \hspace{1cm}(3)

where $M^l$ is the updated occlusion mask in level $l$, $\sigma_\tau$ and $occ$ denotes the sigmoid function and the occlusion network respectively, $\odot$ is the Hadamard product, $[\cdot]$ represents the concatenation operation.

Secondly, we employ a correlation layer [6] to compute matching costs between the warped pyramid feature and the temporary intermediate feature.

$$(C_0^l, C_1^l) = corr(F_{l}^t, F_{0,0}^{l-1}, F_{1,0}^{l-1})$$  \hspace{1cm}(4)

where $corr$ denotes the correlation layer, $C_0^l$ and $C_1^l$ are the learned correlations.

Finally, we fuse the correlations with the warped pyramid features using a gated unit to generate the residual of the flow, which is then used to update the input flows.

$$v_{l-0} = v_{l-0}^{l-1} + \sigma(gate([C_0^l, F_{0,0}^{l-1}, F_{l}^{l-1}, v_{l-0}^{l-1}]))$$  \hspace{1cm}(5)

$$v_{l-1} = v_{l-1}^{l-1} + \sigma(gate([C_1^l, F_{1,0}^{l-1}, F_{l}^{l-1}, v_{l-1}^{l-1}]))$$  \hspace{1cm}(6)
where $T_{gate}$ represents the gated unit, $v^{l}_{t \rightarrow 0}$ and $v^{l}_{t \rightarrow 1}$ are the updated intermediate flows in level $l$.

3) Anchor Frame Interpolation: We use the refined intermediate flows $v_{t \rightarrow 0}$, $v_{t \rightarrow 1}$ and the learned occlusion mask $M$ to generate a temporary source frame $T_t$, called an anchor frame. The anchor frame can be generated by

$$T_t = M \odot \phi_B(v_{t \rightarrow 0}, I_0) + (1 - M) \odot \phi_B(v_{t \rightarrow 1}, I_1), \quad (7)$$

where $\phi_B(\cdot)$ denotes the backward warping function.

C. Deformation Step

As aforementioned, flow-based methods are often jeopardized by the inaccuracy of flow map estimation. Only flow warping on images or features cannot handle the large motions and complex scenarios. To solve this problem, we design a deformation step to compensate for the missing details of the flow step. Its overall structure is shown in Fig. 4(a), which contains a pyramid deformable compensation network (PDCN) and a multi-level frame synthesis network (MFSN). Different from the previous DConv-based VFI methods that detect the complete motions directly, our PDCN reduces the burden of the offset learning by using the learned flow priors in the flow step as guidance. This strategy not only makes the DConv layer more trainable but also helps it mine more detailed information.

1) Pyramid Deformable Compensation Network: PDCN uses a pyramid network and conducts deformable compensation operations on the multi-level pyramid features. To be specific, for the $l$-th level feature $F^l_0$ and $F^l_1$, we first use the learned intermediate flows $v^{l}_{t \rightarrow 0}$ and $v^{l}_{t \rightarrow 1}$ to warp pyramid features and we get the warped feature $F^{l}_{0,w}$ and $F^{l}_{1,w}$. Then a flow guidance deformable compensation layer (FGDCL) is employed to compensate for the missing information of the flow warped features.

The architecture of FGDCL is shown in Fig. 4(b). It presents the compensation process of the flow warped feature $F^{l}_{0,w}$ under the guidance of the flow map $v^{l}_{t \rightarrow 0}$. As shown in the figure, our FGDCL uses the flow warped feature $F^{l}_{0,w}$ and $F^{l}_{1,w}$ as inputs and utilizes a series of “Conv-ReLU” layers to learn the offset residue $\Delta p^{l}_{t \rightarrow 0}$ and the modulation mask $\Delta m^{l}_{t \rightarrow 0}$. Then the offset residues are added by the pre-estimated optical flow and sent to the DConv layer to warp the original feature $F^l_0$. Mathematically, the flow guidance DConv layer can be formulated as

$$\Delta F^l_{0,w}(p) = \sum_{k=1}^{n^2} w(p_k) \cdot F^l_0(p + p_k + v^{l}_{t \rightarrow 0}(p) + \Delta p_k) \cdot \Delta m_k(p), \quad (8)$$

where $v^{l}_{t \rightarrow 0}(p)$ denotes the pre-estimated flow offset at position $p$, $\Delta F^l_{0,w}$ is the compensation details for the flow warped feature $F^l_{0,w}$. For convenience, we omit the superscript $l$ and subscript $t \rightarrow 0$ of $\Delta p_k$ and $\Delta m_k(p)$.

At last, we design a skip connection (SC) layer to enforce the DConv layer to learn the residuals based on the flow warped feature.

$$F^{l}_{0 \rightarrow t} = \Delta F^{l}_{0,w} + F^{l}_{0,w}, \quad (9)$$

where $F^{l}_{0 \rightarrow t}$ denotes the final warped feature from the input feature $F^l_0$. Using the same operations, we get the warped feature $F^{l}_{1 \rightarrow t}$ from the input feature $F^l_1$. Both the two warped features are concatenated together to get the pyramid intermediate feature $F^l_t$ and then they are sent to MFSN together to get the final reconstruction results. In addition, we design a cascaded offset (CO) estimation structure in PDCN, which helps the network to estimate more accurate offsets.

2) Multi-Level Frame Synthesis Network: From PDCN, we have got a pyramid of warped features. In this part, we employ MFSN to fuse the multi-level warped features and reconstruct the final results. GridNet [16] has shown its effectiveness on frame synthesis in various VFI methods [5], [6]. The original version of GridNet has single a input and single output, here we design a modified version that has multi-level inputs and reconstruct multi-level outputs simultaneously. Its structure is shown in Fig. 5. The network employs the pyramid intermediate features $F^l_{0 \rightarrow t}$ as inputs and reconstructs the image.
residuals $\Delta I_l^t$ after the rightmost lateral block. Thus, the final interpolated intermediate frame in the $l$-th level can be obtained by:

$$I_l^t = T_l^t + \Delta I_l^t,$$

(10)

where $T_l^t$ is the anchor frame at the $l$-th level, $\Delta I_l^t$ is the learned image residual between the anchor frame and target frame.

D. Loss Functions

We adopt a two-stage training strategy to train our FGDCN. In the first stage, we train the flow step using the $L_{fstep}$, including a distillation loss and a task-oriented loss. In the second stage, we train the overall network using $L_{total}$, including a pyramid content loss, a pyramid frequency loss, a distillation loss and a census loss.

1) Distillation Loss: The flow distillation loss is employed to supervise the coarse intermediate flows, which can be formulated as:

$$L_{dis} = ||\mathbf{v}_t \rightarrow 0 - v_{l,gt}^t||_1 + ||\mathbf{v}_{t} \rightarrow 1 - v_{l,gt}^t||_1,$$

(11)

where $\mathbf{v}_t \rightarrow 0$ and $\mathbf{v}_t \rightarrow 1$ represent the estimated intermediate flows, $v_{l,gt}^t$ and $v_{l,gt}^t$ are the pseudo labels estimated by Liteflow [38]. Since the pseudo labels are not entirely accurate, we only use them for weak supervision. To be specific, the weak supervision is conducted in a downsampling resolution instead of the full resolution, which avoids some inaccurate guidance from the pseudo label.

2) Task-Oriented Loss: To estimate an accurate intermediate flow, we train the coarse-to-fine flow estimation network in a task-oriented manner.

$$L_{to} = ||I^t_{gt} - \mathbf{T}_l||_1,$$

(12)

where $I^t_{gt}$ is the ground truth image. Hence, the loss function of the flow step is defined as

$$L_{fstep} = L_{to} + \lambda_{dis} L_{dis},$$

(13)

where $\lambda_{dis}$ is the loss weight for $L_{dis}$.

3) Pyramid Content Loss: Following the previous methods, we adopt the $L_1$ loss to supervise the reconstruction contents. Since we reconstruct the multi-level outputs, our pyramid content loss can be represented as

$$L_{pc} = \frac{1}{L} \sum_{l=1}^{L} ||I_l^{t,gt} - I_l^t||_1,$$

(14)

where $L$ is the total level of our pyramid network, $I_l^{t,gt}$ and $I_l^t$ is the ground truth image and the reconstruction image in level $l$.

4) Pyramid Frequency Loss: Existing VFI methods tend to use pixel-wise loss, e.g., $L_1$ loss, to supervise the reconstruction content, which lacks the supervision of global information. Here we employ a frequency loss in the Fourier space to introduce global guidance as opposed to pixel-wise loss due to the nature of the Fourier transform [40]. Our frequency loss is also carried out on different levels:

$$L_{pf} = \frac{1}{L} \sum_{l=1}^{L} ||T_l^{fft}(I_l^{t,gt}) - T_l^{fft}(I_l^t)||_1,$$

(15)

where $T_l^{fft}$ denotes the fast Fourier transform (FFT) that transfers the image signal to the frequency domain. Since the data in the frequency domain is complex data that contains an amplitude component and a phase component, we calculate the loss of each component separately and add them as the total frequency loss.

5) Census Loss: Census loss [41] $L_{sen}$ is robust for illumination changes, which has been widely used in VFI tasks. It is represented as the soft Hamming distance between census-transformed image patches of $I_l^{t,gt}$ and $I_l^t$.

6) Total Loss: According to the above analysis, our full loss function is defined as

$$L_{total} = L_{pc} + \lambda_{pf} L_{pf} + \lambda_{dis} L_{dis} + \lambda_{sen} L_{sen},$$

(16)

where $\lambda_{pf}$ and $\lambda_{sen}$ are loss weights for $L_{pf}$ and $L_{sen}$ respectively. $L_{dis}$ denotes the distillation loss.

IV. EXPERIMENTS

A. Implementation Details

1) Training Details: We implement the proposed algorithm in PyTorch and use Vimeo90K [24] training set to train FGDCN. We use Adam optimizer to optimize the network. The training process contains two stages. In the first stage, we train the flow step using $L_{fstep}$ for $2 \times 10^5$ iterations with batch size 24, patch size $192 \times 192$. The learning rate is set to $1 \times 10^{-4}$. In the second stage, we train the whole model using $L_{total}$ in an end-to-end manner for $4 \times 10^5$ iterations with batch size 32, patch size $128 \times 128$. The learning rate is initially set to $1 \times 10^{-4}$, and decays to $1 \times 10^{-5}$ using a cosine attenuation schedule. We also
attempt to train the entire network directly in an end-to-end manner, but it will lead to slight performance degradation (discuss in the ablation study). The weight coefficients $\lambda_{pf}$, $\lambda_{dis}$ and $\lambda_{sen}$ are 0.1, 0.01 and 1, respectively. For the first stage, training patches are randomly cropped from the training samples and augment them by random flip and time reversal.

2) Model Details: We design two versions of FGDCN, an FGDCN-Large (FGDCN-L) and an FGDCN-Small (FGDCN-S). In FGDCN-L, the channel numbers of the three IFBs are set to 180, 120, and 90. The channel numbers of the three-level pyramid network are set to 96, 48, and 24. The channel numbers of the different level lateral blocks are set to 192, 96, and 48. In FGDCN-S, the channel numbers of the three IFBs are set to 120, 90, and 60. The channel numbers of the three-level pyramid network are set to 96, 48, and 24. The channel numbers of the different level lateral blocks are set to 96, 48, and 24.

B. Evaluation Metrics and Datasets

We evaluate our method on various datasets covering diverse motion scenes.

1) Vimeo90K [24]: The Vimeo90K training set contains 51,312 triplets, where each triplet consists of three consecutive video frames with resolution $448 \times 256$. Its testing set contains 3,782 triplets and their resolution are also $448 \times 256$.

2) UCF101 [42]: It contains videos with a large variety of human actions. There are 379 triplets with a resolution of $256 \times 256$.

3) Middlebury: The Middlebury benchmark is a widely used dataset to evaluate optical flow and VFI methods. Image resolution in this dataset is around $640 \times 480$. we report the average interpolation error (IE) on the OTHER dataset.

4) SNU-FILM [39]: It contains 1,240 triplets of resolutions up to $1280 \times 720$. There are four different settings according to the motion types: Easy, Medium, Hard, and Extreme.

PSNR and SSIM are adopted for quantitative evaluation. Following previous methods [19], we report the average interpolation error (IE) on the Middlebury testing set. A higher PSNR and SSIM, and lower IE indicate better performance.

C. Comparisons With State-of-the-Art Methods

We comprehensively compare our FGDCN-S and FGDCN-L with recent state-of-the-art methods, including DAIN [18], CAIN [39], AdaCof [32], SoftSplat [5], RIFE [15], ABME [6], [6],
TABLE I
QUANTITATIVE RESULTS OF DIFFERENT VFI METHODS

| Method     | Params | Vimeo90K↑ | UCF101↑ | M.B.↓ | SNU-FILM↑ |
|------------|--------|----------|--------|-------|----------|
| DAIN [18]  | 24.0M  | 34.71/0 9756 | 34.99/0 9683 | 2.04  | 39.73/0.9902 | 35.46/0.9780 | 30.17/0.9335 | 30.17/0.9335 |
| CADN [39]  | 42.8M  | 34.65/0 9730 | 34.91/0 9690 | 2.28  | 39.89/0.9900 | 35.61/0.9776 | 29.90/0.9292 | 24.78/0.8507 |
| AdaCoF [32] | 21.8M  | 34.47/0 9730 | 34.90/0 9680 | 2.24  | 39.80/0.9900 | 35.05/0.9754 | 29.46/0.9242 | 24.31/0.8439 |
| SoftSplat [5] | 12.2M  | 36.10/0 9700 | 35.39/0 9520 | -     | -        | -        | -        | -        |
| RIFE [15]  | 9.8M   | 35.62/0 9779 | 35.28/0.9688 | 1.96  | 40.06/0.9907 | 35.75/0.9789 | 30.10/0.9330 | 21.48/0.8534 |
| ABME [6]   | 18.1M  | 36.18/0 9805 | 35.38/0 9698 | 2.01  | 39.59/0.9901 | 35.77/0.9789 | 30.58/0.9364 | 25.42/0.8639 |
| IFRNet-L [7] | 19.7M  | 36.20/0 9808 | 35.42/0 9698 | 1.89  | 40.10/0.9906 | 35.12/0.9797 | 30.63/0.9368 | 25.27/0.8609 |
| VFIformer-S [19] | 17.0M  | 36.38/0.9811 | 35.36/0 9700 | 1.90  | 39.92/0.9905 | 35.97/0.9793 | 30.53/0.9366 | 25.35/0.8634 |
| VFIformer-L [19] | 24.1M  | 36.50/0.9816 | 35.43/0 9700 | 1.82  | 40.13/0.9907 | 36.09/0.9799 | 30.67/0.9378 | 25.43/0.8643 |
| FGDCN-S     | 6.5M   | 36.24/0 9806 | 35.42/0.9698 | 1.94  | 40.31/0.9909 | 36.12/0.9780 | 30.70/0.9376 | 25.51/0.8631 |
| FGDCN-L     | 14.4M  | 36.46/0 9814 | 35.43/0 9701 | 1.86  | 40.40/0.9912 | 36.23/0.9803 | 30.82/0.9382 | 25.58/0.8639 |

The bolded and underlined data represent the best and second-best results, respectively.

1 Most reported comparison results are copied from their original papers. For some missing testing results, we refer to the results reported in IFRNet [7] and VFIformer [19].

We also visually compare the running time, the number of parameters and the model performance of existing SOTA methods in Fig. 6. It can be clearly seen that our methods are more efficient than existing SOTA methods.

Fig. 7 shows the visual comparison of different VFI methods on the Vimeo90K testing sets. Fig. 8 shows the visual comparison of recent SOTA VFI methods on the SNU-film Hard and Extreme testing sets, which contain various scenes with fast motions, including jousting, skateboarding, flying birds and playing soccer. As shown in the figure, other methods fail to make accurate predictions of these motions, while our method successfully predicts the motions and generates more reasonable results. It indicates that our flow plus DConv compensation learning strategy is capable of handling complex and fast motions.

D. Ablation Studies

1) Model Architecture Ablation: In this part, we decompose our model into four sub-networks, i.e., flow estimation network (FEN), flow refinement network (FRN), pyramid deformable compensation network (PDCN), and multi-level frame synthesis network (MFSN). The performance comparisons are shown in Table III. Models 1 and 2 demonstrate the usefulness of the proposed FRN and it improves about 0.5-1 dB PSNR on different testing sets. Model 3 is a pure DConv method that uses PDCN to model the complete motions without flow guidance. It works well on simple data, but poorly on complex scenarios. For a fair comparison, we also design an improved version of Model 3, named Model 3+, by increasing its channel numbers. However, the performance gains are very limited. Model 4 is a pure flow-based method. Model 5 introduces PDCN in model 4, its performance improves 0.1-0.25 dB on different testing sets with only 1.1 M parameters increased. Fig. 9 compares the training curves, the reconstruction results, and the learned flows.

Fig. 8. Visual comparison among existing SOTA VFI methods on the SNU-film Hard and Extreme testing sets.

Inputs (Overlay)  ABME  IFRNet-L  Ours  GT

TABLE II
RUNNING TIME COMPARISON OF RECENT SOTA METHODS

| Method      | 448 × 256 | ABME  | IFRNet-L | VFIformer-S / -L | FGDCN-S / -L |
|-------------|-----------|-------|----------|-----------------|--------------|
| 143ms       | 56ms      | 181ms | 278ms    | 40ms / 67ms     |              |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE III
ABLATION STUDY ON DIFFERENT ARCHITECTURE VARIANTS

| Method | Architecture | Params | Vimeo90K | SNU-FILM |
|--------|--------------|--------|----------|----------|
|        | PEN FRN PDCN MFSN |        | Easy | Medium | Hard | Extreme |
| F-Step | Model 1 ✓✓✓ 4.2M | 34.03/0.9711 | 39.53/0.9900 | 35.16/0.9774 | 29.31/0.9256 | 24.01/0.8382 |
|        | Model 2 ✓✓✓ 5.7M | 35.33/0.9775 | 39.97/0.9906 | 35.46/0.9784 | 29.73/0.9301 | 24.36/0.8483 |
| DConv  | Model 3 ✓✓✓ 8.7M | 35.49/0.9773 | 40.21/0.9917 | 35.42/0.9780 | 29.37/0.9262 | 24.27/0.8454 |
|        | Model 3+ ✓✓✓ 17.6M | 35.75/0.9800 | 40.28/0.9947 | 35.53/0.9815 | 29.65/0.9298 | 24.30/0.8488 |
| D-Step | Model 4 ✓✓✓✓ 13.8M | 36.25/0.9811 | 40.30/0.9911 | 36.08/0.9802 | 30.62/0.9376 | 25.32/0.8640 |
|        | Model 5 ✓✓✓✓ 14.4M | 36.46/0.9814 | 40.40/0.9912 | 36.23/0.9803 | 30.82/0.9382 | 25.58/0.8639 |

Fig. 9. (a) The training curves of different architecture variants. (b) The reconstruction results on different architecture variants. (c) The learned intermediate flow of different architecture variants.

TABLE IV
ABLATION STUDY ON DIFFERENT INTERMEDIATE FLOW ESTIMATION METHODS

| Method       | Flow-Net     | Params | Vimeo90K |
|--------------|--------------|--------|----------|
| Linear Comb. | PWC-Net 9.4M | 34.21/0.9719 |
| Flow Proj.   | PWC-Net 9.4M | 32.31/0.9620 |
| Task-Ori.    | Ours 5.7M    | 34.94/0.9755 |
| Task-Ori.+Dis.| Ours 5.7M   | 35.33/0.9775 |

Fig. 10. Quantitative comparisons of different models. As shown in the figure, model 3+ fails to estimate the accurate motion of hands. Model 4 correctly predicts the motion of hands, but leaves severe artifacts in occluded areas. Model 5 not only captures accurate motion information but also solves the occlusion problem well. These experiments demonstrate the superiority of the proposed FGDCN compared with the pure flow-based and pure DConv-based models. More visual comparisons are shown in Fig. 10.

2) Intermediate Flow Estimation: Estimating accurate intermediate flows is critical for the VFI task. In our method, reliable intermediate flows can better guide the learning process of the subsequent DConv layer. In this part, we investigate the performance of different intermediate flow estimation strategies. In Table IV, we compare four strategies, i.e., linear combination [3], flow projection [23], task-oriented [24] and our distillation plus task-oriented strategy. The first two methods employ PWC-Net [43] as the basic flow estimation network and then predict the intermediate flows using the linear combination and flow projection strategies respectively. The next two methods use our coarse-to-fine flow estimation network to predict the intermediate flow directly. All the models are trained by $2 \times 10^5$ iterations. As listed in the table, the task-oriented strategy is better than the linear combination and flow projection strategies. Besides, the distillation strategy can further improve performance significantly.

We also visually compare the four intermediate flow estimation strategies in Fig. 11. As shown in the figure, the flow projection strategy and the linear combination strategy can approximately estimate the intermediate flows, but the results have severe blur and artifacts at the edge of the objects. The task-oriented strategy directly estimates the intermediate flows with sharp edges. However, due to the lack of supervision from motion priors, the flows may be inaccurate in some cases. Our joint distillation and task-oriented optimization strategy estimate the most accurate intermediate flows.

3) FGDCN Ablation: In this article, we propose a flow guidance deformable compensation layer (FGDCL) to compensate for the missing details of the flow warping. To balance the model performance and complexity, the kernel size and the group number of the DConv layer are set to 3 and 8 in our final model. It means that there are $3 \times 3 \times 8 = 72$ additional offsets and masks will be learned for each position. Here we conduct ablation experiments for the FGDCN. As shown in Fig. 4(b), we discuss the effectiveness of the following three components: flow guidance (FG), flow warped skip connection (SC), and cascade offset (CO). Especially, when we remove the flow guidance layer, the original feature $F_0$ will be replaced by the warped feature $F_{0,w}$. For fast evaluation, we only train each model for $2 \times 10^5$ iterations. The quantitative comparisons are shown in...
Fig. 10. Visual comparison of the pure DConv-based model, pure flow-based model and our FGDCN model.

Fig. 11. Visual comparison of different intermediate flow estimation methods on the Vimeo90K testing set.

Table V: Ablation study on FGDCN

| FG | SC | CO | Vimeo90K | SNU-hard |
|----|----|----|----------|----------|
| ✓  | ✓  | ✓  | 36.12/0.9806 | 30.58/0.9375 |
| ✓  | ✓  | ✓  | 36.16/0.9807 | 30.58/0.9374 |
| ✓  | ✓  | ✓  | 36.15/0.9807 | 30.59/0.9375 |
| ✓  | ✓  | ✓  | 36.20/0.9809 | 30.65/0.9376 |

Table VI: Two-stage training vs. one-stage training

|                  | Vimeo90K  | SNU-hard |
|------------------|-----------|----------|
| One-stage training | 36.31/0.9808 | 30.68/0.9376 |
| Two-stage training | 36.46/0.9814 | 30.82/0.9382 |

Table V, which indicates the effectiveness of each component in FGDCN.

In Fig. 12, we randomly select four testing images to visualize their intermediate components in FGDCN. As shown in the figure, the offset residual focus on the motion areas, especially for the edges of the moving objects, where the optical flows have low fidelity. It indicates that the DConv layer focuses on exploring the missing details of the flow warping. The learned modulation mask of the DConv layer is similar to the occlusion mask. It is more smooth since it is the average value of multiple masks. (g) and (h) represent the learned high-frequency details of the DConv layer and the deformation step, respectively.

4) Two-Stage Training Vs. One-Stage Training: In this article, we use a two-stage training strategy to train FGDCN. In the first stage, we train the flow step using $L_{\text{flow}}$ for $2 \times 10^5$ iterations. In the second stage, we train the whole model using $L_{\text{total}}$ in an end-to-end manner for $4 \times 10^5$ iterations. In this part, we compare the two-stage training strategy with the one-stage training strategy. For the one-stage training, we train the entire network using $L_{\text{total}}$ in an end-to-end manner for $6 \times 10^5$ iterations. As listed in Table VI, two-stage training outperforms one-stage training by 0.15 dB PSNR on the Vimeo90K testing set. We analyze that the pre-training of the flow step provides accurate optical flow for the DConv layer, which can better guide the learning of the DConv layer and enforce it to handle hard examples.

5) Loss Function Ablation: In this article, we propose to jointly reconstruct the multi-level images to better optimize the pyramid synthesis network. Table VII compares the quantitative
results under different combinations of the proposed loss functions. Followed by [19], we regard the single-level content loss $L_c$, the distillation loss $L_d$ and the sensus loss $L_s$ as a baseline. As listed in Table VII, it can be seen that our multi-level content loss $L_c^{p}$ improves PSNR by 0.04 dB on Vimeo90K compared with the single-level content loss. When introducing the multi-level frequency loss $L_f^{p}$, the PSNR further improves by 0.07 dB on Vimeo90K. It demonstrates that the proposed pyramid image and frequency losses can effectively improve the performance of our model.

V. CONCLUSION

In this article, we propose a flow guidance deformable compensation network (FGDCN) for video frame interpolation. To the best of our knowledge, this is the first work that combines the motion-based with the DConv-based methods for solving the VFI problem. To be specific, our method decomposes the frame sampling process into two steps, a flow step that is used to estimate the intermediate flow in a coarse-to-fine manner, and a deformation step that is used to compensate for the missing details of the flow step. This structure successfully establishes the relations between the two methods and combines the strengths of both. Besides, pyramid reconstruction losses are employed to supervise the model in both the image and frequency domains. Both quantitative and qualitative results demonstrate the advantage of our method over the existing state-of-the-art methods.

Our method also has limitations. Although our FGDCN is more efficient (about 25FPS on RTX3090 GPU) than other existing SOTA methods, it still cannot satisfy the need for real-time VFI. In the future research, we will work on developing its real-time version.

REFERENCES

[1] M. Usman et al., “Frame interpolation for cloud-based mobile video streaming,” IEEE Trans. Multimedia, vol. 18, no. 5, pp. 831–839, May 2016.
[2] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, “Video frame synthesis using deep voxel flow,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 4473–4481.
[3] H. Jiang et al., “Super SloMo: High quality estimation of multiple intermediate frames for video interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9000–9008.
[4] Y. Liu et al., “Enhanced quadratic video interpolation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 41–56.
[5] S. Niklaus and F. Liu, “Softmax splatting for video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5437–5446.
[6] J. Park, C. Lee, and C.-S. Kim, “Asymmetric bilateral motion estimation for video frame interpolation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 14539–14548.
[7] L. Kong et al., “IFRNe: Intermediate feature refine network for efficient frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 1969–1978.
[8] Z. Shi, X. Liu, K. Shi, L. Dai, and J. Chen, “Video frame interpolation via generalized deformable convolution,” IEEE Trans. Multimedia, vol. 24, pp. 426–439, 2022.
[9] S. Niklaus, L. Mai, and F. Liu, “Video frame interpolation via adaptive separable convolution,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 261–270.
[10] S. Niklaus, L. Mai, and F. Liu, “Video frame interpolation via adaptive convolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 670–679.
[11] J. Dai et al., “Deformable convolutional networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2017, pp. 764–773.
[12] S. Gui, C. Wang, Q. Chen, and D. Tao, “FeatureFlow: Robust video interpolation via structure-to-texture generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 14004–14013.
[13] Z. Chen, R. Wang, H. Liu, and Y. Wang, “PDWN: Pyramid deformable warping network for video interpolation,” IEEE Open J. Signal Process., vol. 2, pp. 413–424, 2021.
K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, “Understanding deformable alignment in video super-resolution,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 973–981.

Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, “RIFE: Real-time intermediate flow estimation for video frame interpolation,” 2021, arXiv:2011.06294.

S. Niklaus and F. Liu, “Context-aware synthesis for video frame interpolation,” in Proc. IEEE/CVF Comput. Vis. Pattern Recognit., 2018, pp. 1701–1710.

F. Reda et al., “Unsupervised video interpolation using cycle consistency,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 892–900.

W. Bao et al., “Depth-aware video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3703–3712.

L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, “Video frame interpolation with transformer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3532–3542.

Z. Shi, X. Xu, X. Liu, J. Chen, and M.-H. Yang, “Video frame interpolation transformer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 17482–17491.

D. Danier, F. Zhang, and D. Bull, “ST-MFNet: A spatio-temporal multi-flow network for frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3521–3531.

P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko, “Many-to-many splatting for efficient video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3535–3562.

W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, “MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 3, pp. 933–948, Mar. 2021.

T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video enhancement with task-oriented flow,” Int. J. Comput. Vis., vol. 127, pp. 1106–1125, 2019.

Z. Zhu, H. Hu, S. Lin, and J. Dai, “Deformable ConVNeTs v2: More deformable, better results,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9308–9316.

G. Bertasius, L. Torresani, and J. Shi, “Object detection in video with spatiotemporal sampling networks,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 331–346.

K.-N. Mac et al., “Learning motion in feature space: Locally-consistent deformable convolution networks for fine-grained action detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6282–6291.

L. Deng et al., “Restricted deformable convolution-based road scene semantic segmentation using surround view cameras,” IEEE Trans. Syst. Soc., vol. 21, no. 10, pp. 4350–4362, Oct. 2020.

Y. Tian, Y. Zhang, Y. Fu, and C. Xu, “TDAN: Temporally-deformable alignment network for video super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3360–3369.

X. Wang, K. C. Chan, K. Yu, C. Dong, and C. C. Loy, “EDVR: Video restoration with enhanced deformable convolutional networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2019, pp. 1954–1963.

X. Yang et al., “Zooming slow-mo: Fast and accurate one-stage spacetime video super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3370–3379.

H. Lee et al., “AdaCoF: Adaptive collaboration of flows for video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5316–5325.

X. Cheng and Z. Chen, “Multiple video frame interpolation via enhanced deformable separable convolution,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 10, pp. 7029–7045, Oct. 2022.

J. Liu, Y. Huang, and L. Wang, “FDAN: Flow-guided deformable alignment network for video super-resolution,” 2021, arXiv:2012.05640.

K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, “BasicVSR++: Improving video super-resolution with enhanced propagation and alignment,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 5972–5981.

Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng, “Towards an end-to-end framework for flow-guided video inpainting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 17562–17571.

Z. Luo et al., “BSRT: Improving burst super-resolution with swin transformer and flow-guided deformable alignment,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 998–1008.

T.-W. Hui, X. Tang, and C. C. Loy, “LiteFlowNet: A lightweight convolutional neural network for optical flow estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8981–8989.

M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, “Channel attention is all you need for video frame interpolation,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 10663–10671.

D. Fuoli, L. V. Gool, and R. Timofte, “Fourier space losses for efficient perceptual image super-resolution,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 2360–2369.

S. Meister, J. Hur, and S. Roth, “UnFlow: Unsupervised learning of optical flow with a bidirectional census loss,” in Proc. 32nd AAAI Conf. Artif. Intell. 30th Innov. Appl. Intell. Conf. 8th AAAI Symp. Educ. Adv. Intell. Intell., 2018, pp. 7251–7259.

K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012, arXiv:1211.0402.

D. San, X. Yang, M.-Y. Liu, and J. Kautz, “PW-CNet: CNNs for optical flow using pyramid, warping, and cost volume,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8934–8943.

Pengcheng Lei received the B.E. degree from the North China University of Water Resources and Electric Power, Zhengzhou, China, in 2018. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include image and video super-resolution.

Faming Fang received the Ph.D. degree in computer science from East China Normal University, Shanghai, China, in 2013. He is currently a Professor with the Department of Computer Science and Technology, East China Normal University, Shanghai. His main research interests include image processing using the variational methods, PDEs, and deep learning methods.

Tieyong Zeng received the B.S. degree from Peking University, Beijing, China, in 2000, the M.S. degree from École Polytechnique, Palaiseau, France, in 2004, and the Ph.D. degree from the Université de Paris XIII, Paris, France, in 2007. He is currently an Associate Professor with the Department of Mathematics, The Chinese University of Hong Kong, Hong Kong. From 2007 to 2008, he was a Postdoctoral Researcher with the ENS de Cachan and from 2008 to 2018, he was an Assistant/Associate Professor with Hong Kong Baptist University, Hong Kong. His research interests include image processing, machine learning, and scientific computing.

Guixu Zhang received the Ph.D. degree from the Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, China, in 1998. He is currently a Professor with the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include hyperspectral remote sensing, image processing, and artificial intelligence.