Abstract—Video frame interpolation can up-convert the frame rate and enhance the video quality. In recent years, although interpolation performance has achieved great success, image blur usually occurs at object boundaries owing to the large motion. It has been a long-standing problem and has not been addressed yet. In this brief, we propose to reduce the image blur and get the clear shape of objects by preserving the edges in the interpolated frames. To this end, the proposed edge-aware network (EA-Net) integrates the edge information into the frame interpolation task. It follows an end-to-end architecture and can be separated into two stages, i.e., edge-guided flow estimation and edge-protected frame synthesis. Specifically, in the flow estimation stage, three edge-aware mechanisms are developed to emphasize the frame edges in estimating flow maps, so that the edge maps are taken as auxiliary information to provide more guidance to boost the flow accuracy. In the frame synthesis stage, the flow refinement module is designed to refine the flow map, and the attention module is carried out to adaptively focus on the bidirectional flow maps when synthesizing the intermediate frames. Furthermore, the frame and edge discriminators are adopted to conduct the adversarial training strategy, so as to enhance the reality and clarity of synthesized frames. Experiments on three benchmarks, including Vimeo90k, UCF101 for single-frame interpolation, and Adobe240-fps for multi-frame interpolation, have demonstrated the superiority of the proposed EA-Net for the video frame interpolation task.

Index Terms—Adversarial training, edge-aware, flow estimation, video frame interpolation.

I. INTRODUCTION

With the popularity of smartphones and digital cameras, people are full of desire to record and share their memorable moments in daily lives [1], [2]. However, the quality of user-captured videos is usually uneven, which impacts the visual effect. In this case, there is an urgent demand for automatic techniques to enhance the video quality [3], [4]. Video frame interpolation is such an effective technique rising for video quality enhancement.

Video frame interpolation can up-convert the frame rate of videos by interpolating intermediate frames between consecutive frames [5], [6], e.g., from 30 to 240 frames/s, so that the motions in the video stream become smoother. By taking this advantage, it can be applied to different video processing tasks, such as animation production, high-speed photography, and slow motion generation. Overall, video frame interpolation is a fundamental task in computer vision and graphics with great application potentials.

Generally, there is a stable development of video frame interpolation in [7] and [8]. Traditional approaches are mainly composed of two parts, i.e., motion estimation and frame synthesis [9], [10]. They estimate the object motion by computing the optical flow between consecutive frames and synthesize the intermediate frames by warping the two frames according to the computed flow map.

Recently, video frame interpolation has achieved great progress benefiting from the convolutional neural networks (CNNs) [11]–[13]. The CNN-based approaches share similar architectures with the traditional ones. The main differences lie in that CNNs are adopted for flow estimation and frame synthesis by taking advantages of their visual feature capturing ability. Furthermore, flow estimation and frame synthesis parts are combined together in CNN-based approaches for end-to-end training.

Although tremendous progress has been made in video frame interpolation, it is still a challenging task for high-quality frame generation. In general, large motion and occlusions of objects tend to make frame blur and artifacts. The existing approaches try to address this problem using more advanced flow estimators pre-trained on large-scale datasets [14], [15]. Moreover, depth information and object masks are also used to predict the occlusions [16], [17]. By integrating more information, the performance is improved. However, these approaches are in more complex architectures and require more pretrained models and extra annotated datasets, which increases the training difficulty and makes them hard to operate in the wild.

Practically, the frame blur and artifacts caused by large motion and occlusion usually occur at object boundaries. Considering that frame edges draw the boundaries of different objects, we propose to improve the quality of interpolated frames by preserving the frame edges explicitly. Motivated by this, the edge-aware network (EA-Net) is developed in this brief for high-quality video frame interpolation. It follows an end-to-end architecture and can be separated into two stages, i.e., edge-guided flow estimation and edge-protected frame synthesis. In the flow estimation stage, the frame and its extracted edge map are integrated together for flow estimation, so that the flow maps are estimated under the guidance of edge information. Particularly, three edge-aware mechanisms are developed for frame and edge integration, including concatenation, augmentation, and two-stream, to analyze their influence on the performance. In the frame synthesis stage, a flow refinement module is designed to refine and interpolate the flow map. Then, an attention module is used to adaptively attend to the bidirectional flow map when synthesizing the intermediate frames. Besides, the adversarial training strategy is conducted by adopting the frame and edge discriminators to further enhance the reality and preserve the boundary of synthesized frames. The experiments are conducted on three benchmark datasets for both the single- and multiple-frame interpolation tasks. The results demonstrate that the proposed EA-Net can significantly improve the performance by integrating the edge information explicitly. Moreover, it achieves comparable results with state-of-the-arts in a much compact architecture and requires no pretrained models and extra annotated data.

In this brief, the contributions of the proposed EA-Net are summarized into three folds.

1) The edge information is integrated into the video frame interpolation task by adding three simple and effective edge-aware mechanisms, which can reduce the interference caused by large motion and occlusions.

2) The flow attention module is designed to adaptively attend to the bidirectional flow maps, so that more accurate motion information is used for frame synthesis.
3) The adversarial training strategy is developed by adopting the frame and edge discriminators, which can enhance the reality of frames and the clarity of object boundaries.

II. RELATED WORKS

A. Phase-Based Video Frame Interpolation

Motion estimation is the key step in the video frame interpolation task [18]. Phase-based approaches estimate the motion in the videos by computing the phase shift information between frames. This kind of approaches are developed based on the assumption that video motion is captured by the phase shift of pixel color, which have shown promising results in view expansion [19], [20] and motion magnification [21], [22]. However, conventional phased-based approaches can only deal with small range of motion information. To remedy this problem, Meyer et al. [23] develop a multiscale pyramid model and adopt a coarse-to-fine strategy to extend phase shift limitation, where the phase information is propagated across different levels with a bounded shift correction method. Furthermore, the PhaseNet is proposed to integrate deep learning to phase difference computation [24], which is more robust to handle different scenarios. Besides, Fahim et al. [25] use the edge-preserving guided filtering before computing the phase difference, so that the object boundaries in the frames are protected.

B. Kernel-Based Video Frame Interpolation

The kernel-based approaches use convolutional neural networks to model video motion estimation and frame interpolation into an end-to-end architecture [26]–[28]. Specifically, Niklaus et al. [29] develop an adaptive convolution model to estimate convolution kernels. Naturally, large kernels are used for large motion, and vice versa. However, large kernels will increase the computation memory significantly. To reduce this problem, a separable-adaptive convolution model is proposed with the assumption that the 2-D kernel can be separated into a pair of 1-D kernels [30]. In this case, the memory consumption is reduced. However, the main drawback of the kernel-based approaches is that they cannot deal with motion larger than kernel size. Facing this problem, Bao et al. [17] propose MEMC-Net to integrate motion kernel and optical flow together for video frame interpolation.

C. Flow-Based Video Frame Interpolation

The flow-based approaches are most popular in video frame interpolation. This kind of approaches use optical flow to represent video motion [31], [32]. Compared with the aforementioned two kinds of approaches, the main advantages of flow-based approaches are that they can generate multiple intermediate frames [33], [34]. However, the quality of interpolated frames heavily depends on the accuracy of optical flow. Actually, optical flow computation is a challenging problem in computer vision, since it needs to estimate pixelwise correspondence. In this case, conventional optical flow computation methods suffer from serious frame blurriness and artifacts [35], [36].

To improve the flow accuracy and interpolation quality, Liu et al. [37] propose a deep voxel flow method to learn optical flow with 3-D convolutional neural networks, and then generate intermediate frames by trilinear sampling. Jiang et al. [33] adopt the U-Net architecture for bidirectional flow computation and multiple frame interpolation. Niklaus et al. propose a context-aware video frame interpolation approach, which adopts the PWC-Net [15] as the flow estimator and generates the intermediate frame with a spatial warping layer. Furthermore, Bao et al. [16] develop a depth-aware approach that integrates the flow, depth, context, and kernel together and interpolates frames with an adaptive warping layer. However, depth estimation is also a difficult task and needs pre-training on extra datasets. Lee et al. [11] present adaptive collaboration of flows to handle more complex video motion and boost the performance. Although flow-based approaches have made tremendous progress in video frame interpolation, they still suffer from object boundary blurriness due to the large motion between frames. In this case, we propose a flow-based EA-Net to preserve object boundaries in flow estimation and frame interpolation.

Overall, although great success has been achieved in recent years, frame blur caused by large motion still occurs at object boundaries. Few approaches realize the importance of edge preserving in frame interpolation. Practically, its effectiveness has been proven in relevant tasks [38]–[40]. Inspired by this, we propose an EA-Net for video interpolation in this brief.

III. EDGE-AWARE NETWORK FOR VIDEO INTERPOLATION

In this brief, we propose an EA-Net for the video frame interpolation task. As depicted in Fig. 1, it consists of two stages, i.e., edge-guided flow estimation and edge-protected frame synthesis. Specifically, the edge-guided flow estimation part develops three mechanisms, including edge augmentation, edge concatenation, and two-stream, to estimate the video motion under the guidance of object boundaries. The edge-protected frame synthesis part adopts a refinement module to refine the flow maps and an attention module to regulate the weights of forward and backward flow in frame synthesis. Besides, the frame and edge discriminators are adopted to conduct adversarial learning. The detailed architecture of the EA-Net is introduced in the following subsections successively.

A. Edge-Guided Flow Estimation

To develop an end-to-end architecture, we adopt U-Net [41] as the backbone to compute optical flow between two consecutive frames, rather than use the existing flow CNNs, such as PWC-Net [15] and FlowNet [14], since they require pre-training on extra datasets. Specifically, U-Net is originally proposed for the medical image segmentation task, which is composed of fully convolutional layers and follows the encoder–decoder architecture. It is quite suitable for the image-to-image task, including image segmentation, image generation, and flow estimation. In our work, U-Net is used to compute the video motion from scratch. It should be noted that the flow map is learned in an unsupervised manner and no extra annotated datasets are required in this process.

Actually, U-Net has been modified as a general network with specific structures for different tasks. In our work, the U-Net adopted for flow estimation is composed of an encoder and a decoder, where the encoder is used to extract the high-level frame feature, and the decoder is used to estimate the motion information given the encoded frame features as input. The encoder contains six down-convolution blocks. Specifically, the filter sizes in the first two layers are set as $7 \times 7$ and $5 \times 5$, since large receptive fields are helpful to capture long-range motion, and the last layers are all with $3 \times 3$ filters. Each block consists of two convolutional layers with Leaky ReLU as the activation functions. Except the last block, they are followed by the $2 \times 2$ max pooling layer to down-sample the feature map. The decoder consists of five up-convolution blocks, which has symmetric structures with the encoder. Specifically, in each block, the only difference is that the pooling layer is replaced by the up-sampling layer, so that the feature maps are up-sampled to the same size with the input frame. Finally, given the consecutive frames as input, the U-Net can generate the bidirectional flow maps correspondingly.
Generally, the accuracy of flow map is essential to the performance of frame interpolation. To better preserve the frame details, we pay special attention to motion at object boundaries. In this case, edge-aware mechanisms are developed in the flow estimation process. First, given frame \( I \) as input, the edge map is generated by the canny edge detection algorithm [42], denoted as \( E \). Based on this, the edge-aware mechanisms are developed as follows.

1) **Edge Augmentation**: The object boundaries are emphasized in the original frame by augmenting the pixel values at edges. It is formulated as follows:

\[
I_{\text{aug}} = \frac{1}{2} (I + I \odot E)
\]

where \( \odot \) stands for the pixelwise multiplication. The above equation means the nonedge pixels are depressed in the flow estimation process, and the edge pixels are augmented correspondingly.

2) **Edge Concatenation**: The original frame and the extracted edge map are concatenated as a six-channel input image. It is formulated as

\[
I_{\text{con}} = [I; I \odot E]
\]

where \([; ;] \) denotes the channelwise concatenation operation. In this case, the number of input channels of U-Net is changed according to \( I_{\text{con}} \).

3) **Two-Stream**: A two-stream structure is developed for U-Net, where the flow is estimated by the frames and their edge maps jointly, denoted as

\[
F = \frac{1}{2} (F^I + F^E)
\]

where \( F^I \) and \( F^E \) stand for the flow map computed by the frame and its edge map, respectively. \( F \) is the finally generated flow map in the two-stream structure.

**B. Edge-Protected Frame Synthesis**

After the flow map is computed, the intermediate frames can be simply synthesized. Specifically, given two consecutive frames \( I_0 \) and \( I_1 \), the intermediate frame \( I_t \) can be synthesized by

\[
I_t = \text{warp}(I_0, F_{t \rightarrow 0}) \quad \text{or} \quad I_t = \text{warp}(I_1, F_{t \rightarrow 1})
\]

where the bilinear interpolation is used as the warping function, denoted as \( \text{warp}() \). \( F_{t \rightarrow 0} \) and \( F_{t \rightarrow 1} \) are the interpolated flow map. Based on the assumption that the objects are moving uniformly in a small time interval, the intermediate flow maps are computed by

\[
F_{t \rightarrow 0} = t F_{t \rightarrow 0} = -t F_{0 \rightarrow t}
\]

\[
F_{t \rightarrow 1} = -(1-t) F_{1 \rightarrow 0} = (1-t) F_{0 \rightarrow t}.
\]

The above describes a simple strategy for intermediate frame synthesis. However, there are remaining two serious problems that can affect the performance.

1) The flow map is not precise enough to synthesize high-quality intermediate frames, since no supervision is provided in the flow estimator. In addition, the linear model in (5) and (6) works well when the flow changes smoothly, but it is not suitable for other situations, especially at object boundaries.

2) There are multiple ways to synthesize the intermediate frames, as depicted in (4), since both the forward and backward flow maps are provided. It is hard to decide which one is better. Practically, the bidirectional flow maps are both important to synthesize the intermediate frame.

To remedy the above problems, another U-Net network is used to refine the flow map and synthesize the intermediate frames. It is cascaded after the flow estimator, and they share similar structures. It takes the original frames \( I_0 \) and \( I_1 \), the estimated flow map \( F_{t \rightarrow 0} \) and \( F_{0 \rightarrow t} \), the interpolated flow map \( F_{t \rightarrow 0} \) and \( F_{1 \rightarrow 0} \), and the warped frames \( \text{warp}(I_0, F_{t \rightarrow 0}) \) and \( \text{warp}(I_1, F_{t \rightarrow 1}) \) as inputs. The outputs have two branches, the first is the refined flow map \( F'_{t \rightarrow 0} \) and \( F'_{1 \rightarrow 0} \). The second are the attention maps \( A_t \) and \( A_0 \), which balance the weights of the forward and backward flow maps in synthesizing the intermediate frame, respectively. Finally, the synthesis of the intermediate frame is formulated as

\[
I_t = A_0 \text{warp}(I_0, F'_{t \rightarrow 0}) + A_1 \text{warp}(I_1, F'_{t \rightarrow 1})
\]
where the size of $A_0$ and $A_1$ is the same as the shape of frames, and
\[ A_0 + A_1 = 1. \]  

(8)

After the interpolated frame is synthesized, the discriminator is developed to further enhance the reality and clarity of the intermediate frames. The targets of the discriminator are to: 1) classify realistic frames from synthesized frames and 2) classify realistic edge maps from synthesized edge maps. In this case, the discriminator has two branches, i.e., the frame discriminator and the edge discriminator. In this brief, each discriminator is composed of four convolutional layers and one sigmoid layer as the last. In each convolutional layer, the filter size is four with stride 2, the initial output channel is 64 and doubles as the layer deepens, and Leaky ReLU is used as the activation function. Besides, the batch normalization layer is stitched after each convolutional layer, which has shown its superiority in reducing the overfitting problem.

C. Optimization

The final loss function of the proposed EA-Net contains three terms, which are formulated as follows:

\[ \text{loss} = l_{\text{syn}} + l_{\text{flow}} + l_{\text{adv}} \]  

(9)

where $l_{\text{syn}}$ is the synthesis loss, $l_{\text{flow}}$ denotes the optical flow loss, and $l_{\text{adv}}$ is the adversarial loss. The details are described as follows.

1) The synthesis loss $l_{\text{syn}}$ measures the similarity of the synthesized frame and the ground truth. In this brief, the similarity is determined by the $L_1$ norm, which has been proven to be more effective to the interpolation task. It is formulated as

\[ l_{\text{syn}} = \| I_0 - I_t \|_1 \]  

(10)

where $I_0$ and $I_t$ stand for the synthesized frame and the ground truth, respectively.

2) The flow loss $l_{\text{flow}}$ measures the performance of the estimated flow map. Considering that the annotated flow map is not available, the results of warping frames via the estimated flow map are utilized to evaluate the performance. It is formulated as

\[ l_{\text{flow}} = \| J_0 \rightarrow \text{warp}(I_0, F_{0 \rightarrow t}) \|_1 + \| I_t \rightarrow \text{warp}(I_0, F_{t \rightarrow 0}) \|_1. \]  

(11)

3) The adversarial loss $l_{\text{adv}}$ can further constrain the realistic appearance and details of the intermediate frame. It consists of two terms, i.e., the frame loss term and the edge loss term

\[ l_{\text{adv}} = D_I(I_t) - D_I(I_0) \]

\[ l_{\text{adv}} = D_E(E_t) - D_E(E_0) \]

\[ l_{\text{adv}} = l_{\text{adv}}^I + l_{\text{adv}}^E \]  

(12)

where $D_I$ and $D_E$ represents the frame discriminator and the edge discriminator, respectively.

The proposed EA-Net is conducted on the deep learning platform of PyTorch. In the training procedure, it is optimized with the Adam optimizer, where the learning rate is initialized as 1e-4, and the decay rate is 0.1 with the milestone of 100 epochs. Practically, convergence can be reached in 500 epochs.

B. Results on Single-Frame Interpolation

In this brief, the single-frame interpolation experiment is conducted on the Vimeo90K and UCF101 datasets. In this section, the proposed EA-Net is first compared with several state-of-the-arts to show its superiority, and second the ablation study is carried out to verify the effectiveness of each component.

1) Comparison With State-of-the-Arts: Table I shows the results of different approaches, including phase-based, kernel-based, and flow-based approaches. Phase shift is one of the typical phase-based approaches, which proposes a multiscale pyramid strategy to enhance the capability to deal with large frame motion. Recently, with the rapid development of deep learning, traditional phase-based approaches have been surpassed by kernel-based and flow-based approaches. Specifically, AdaConv develops a spatially adaptive kernel learning method, which can automatically adjust the kernel size according to the estimated local motion. SepConv is proposed based on AdaConv by separating 2-D kernels into pairs of 1-D kernels. In this case, the training difficulty in large motion is reduced, so that SepConv performs better.

| Approaches | Vimeo90K | UCF101 |
|------------|----------|--------|
|            | PSNR     | SSIM   | PSNR     | SSIM   |
| AdaConv [30] | 33.35    | 0.9584 | 34.69    | 0.9653 |
| SepConv [27] | 35.45    | 0.9674 | 32.40    | 0.9543 |
| SepConv_L [29] | 35.79    | 0.9702 | 34.78    | 0.9699 |
| Vgg [48] | 31.95    | 0.9601 | 33.67    | 0.9633 |
| Vgg [49] | 32.05    | 0.9622 | 33.71    | 0.9633 |
| Vgg [51] | 31.54    | 0.9462 | 34.12    | 0.9631 |
| Vgg [52] | 35.65    | 0.9688 | 34.54    | 0.9663 |
| Vgg [53] | 32.76    | 0.9660 | 34.20    | 0.9707 |
| Vgg [54] | 33.12    | 0.9690 | 35.11    | 0.9654 |
| Vgg [55] | 32.09    | 0.9640 | 34.29    | 0.9632 |
| Vgg [56] | 33.30    | 0.9739 | 34.96    | 0.9652 |
| Vgg [57] | 34.73    | 0.9756 | 34.22    | 0.9683 |

IV. EXPERIMENT

A. Experimental Setup

1) Datasets: The Vimeo90K dataset [43] contains 15 K video clips downloaded from vimeo.com. These video clips are organized as 91 701 frame triplets with a fixed resolution of 448 × 256. They are used to perform the single-frame interpolation task. The training set is composed of 51 313 frame triplets, the test set contains 3782 triplets, and the remaining are used for validation. The UCF101 dataset [44] is a comprehensive video dataset with a variety of human actions. Following existing protocols, the training process is not conducted on UCF101, and the model trained on Vimeo90K is adopted for evaluation directly. Practically, 379 triplets are used for test. They are in the resolution of 256 × 256.

The Adobe240-fps dataset [45] is composed of 133 video clips with the frame rate of 240 frames/s. These clips contain 79768 frames totally with the resolution of 1280 × 720. In this brief, the Adobe240-fps dataset is used for multiframe interpolation. The frames from 112 video clips are used for training, 13 clips for validation, and the remaining eight clips for test.

2) Implementation Details: The proposed EA-Net is trained from scratch with the deep learning platform of PyTorch, on the device of Nvidia GeForce RTX 3090. The training batch size is set as 8. Adam Optimizer is adopted as the optimizer with the initial learning rate as 1e-4. The decay rate is set as 0.1, with 50 and 100 epochs as milestones. Practically, 200 epochs are totally required to reach convergence.

3) Evaluation Metrics: Following existing protocols, two metrics are used for the evaluation of the video frame interpolation task, including Structural SIMilarity (SSIM) and peak signal-to-noise ratio (PSNR). They are typical metrics designed for measuring the similarity of two frames. The metrics are positively correlated with the performance.
than EA-Net, including both SepConv-\textit{L}_1 and SepConv-\textit{L}_2. The proposed EA-Net is developed based on the flow-based architecture. Better performance has demonstrated the superiority of EA-Net in the single-frame interpolation task.

The flow-based approaches are current mainstreams in video frame interpolation. Recently, a variety of deep convolutional networks are proposed for optical flow estimation. SpyNet proposes a spatial pyramid network to estimate the optical flow. Epicflow has recognized the importance of edge information for flow estimation and proposes a sparse-to-dense flow computation scheme, which is more robust to motion boundaries. Better performance of Epicflow than SpyNet indicates the necessity of edge information in video frame interpolation. Our EA-Net outperforms Epicflow significantly. It has verified the superiority of our edge-guided flow estimation and edge-protected frame synthesis. TOFlow is the work constructing the Vimeo90K dataset. It proposes a task-oriented flow computation method, where the optical flow is specially designed for the video interpolation task, so that it gets better performance than previous general approaches, including SpyNet, Epicflow, and DVF.

Furthermore, Super SloMo develops a flow interpolation module to synthesize the intermediate frame. The flow refinement module of EA-Net follows similar structures, but better performance of EA-Net has verified the advantages to consider edge information in flow computation. CyclicGen introduces the cycle consistency loss and uses the holistically nested edge detection (HED) algorithm [49] to integrate edge information to synthesize intermediate frames, which requires extra annotations of edge information. However, our EA-Net can perform comparable with CyclicGen on UCF101 and perform significantly better on Vimeo90K, even without extra annotations. To further promote the performance, IM-Net proposes the multiscale architecture for motion estimation, which can reduce the difficulty in estimating large motion. The edge-guided flow estimation module in EA-Net is an alternative way to address this problem, and the results on the Vimeo90K dataset have verified its superiority.

MEMC-Net and DAIN are two comprehensive approaches. They are both combined with several branches to compute different frame information, including motion, context, kernel, depth, and mask. In this case, they achieve the state-of-the-art on the single-frame interpolation task. However, our EA-Net is an end-to-end approach, which is trained from scratch and does not require any pre-trained feature extractors, depth estimators, etc. Although EA-Net is a simple architecture, we can see that it performs comparable with MEMC-Net and DAIN.

To better compare the results, the intermediate frames synthesized by different approaches are displayed in Fig. 2. Particularly, the results are selected from frames with large motion, like horse riding and parade. From the zoom-in detail of the synthesized frames, it can be clearly observed that EA-Net can better preserve object boundaries. Overall, the results in Fig. 2 and Table I can jointly demonstrate the superiority of the proposed EA-Net in the single-frame interpolation task.

2) Ablation Studies: Table II presents the results of ablation studies on the Vimeo90K and UCF101 datasets. The contributions of three components of EA-Net are analyzed, including the edge-aware mechanism, the attention module, and the discriminator. For the edge-aware mechanism, the results of four baselines are provided. It can be observed that EA-Net without the edge-aware mechanism performs worse than the other three, which meets our motivation and verifies the necessity to integrate edge information for flow estimation. To better understand the results, the examples of estimated flow maps are displayed in Fig. 3. Specifically, in each example, the second row depicts the flow maps estimated by EA-Net without the edge-aware mechanism. The third row is obtained by EA-Net with the edge-aware mechanism. The edge augmentation operation. It can be seen that the accuracy of flow maps are promoted by the edge-aware mechanism, especially at the boundaries. That is why EA-Net with edge-aware mechanisms can outperform the one not. Besides, it can be seen from Table II that EA-Net performs comparably with the three edge-aware mechanisms, which shows its robustness. Closely, we can see that the edge augmentation operation performs slightly better than the other two. Thus, it is used as the default edge-aware mechanism for other comparisons.

Table II presents the comparison of EA-Net with or without the attention module. It should be emphasized that the EA-Net without attention module means the values in \textit{A}_0 and \textit{A}_1 are all 0.5. That is to say, the warping results from the forward and backward flow maps contributed equally to the final interpolated frames. It can be clearly seen that the attention module can significantly boost the performance. To better understand the results, the attention maps.
are displayed in Fig. 4, and we can see that the attention weights distribute evenly in the background and are sensitive to object boundaries obviously, which also illustrates the effectiveness of the proposed edge-aware mechanisms. In this case, EA-Net can better preserve object boundaries.

Table II shows the comparison of the effects of the edge and frame discriminators on interpolation performance. We can see that the performance gets worse by removing each discriminator. It has quantitatively verified the contributions of the proposed edge and frame discriminators to frame interpolation performance. Overall, the ablation studies in Table II have verified the effectiveness of the edge-aware mechanisms in flow estimation, the attention module in frame synthesis, and the discriminators in improving the reality and clarity of synthesized frames.

C. Results on Multiframe Interpolation

In this brief, the multiframe interpolation experiment is conducted on the Adobe240-fps dataset. The results of several state-of-the-arts are compared, and the ablation studies are analyzed successively.

1) Comparison With State-of-the-Arts: Table III shows the results of several state-of-the-arts on Adobe240-fps. It should be noted that single-frame interpolation is a special case of the frame interpolation task, where the number of intermediate frames is fixed as one, and the frame rate can only be doubled. As a result, most compared approaches in the single-frame interpolation task cannot be generalized to the multiframe interpolation task directly, where the arbitrary-time intermediate frame synthesis and flow interpolation are required. In this case, only those approaches that are applicable to the multiframe interpolation task are compared in Table III.

Four typical multiframe interpolation approaches are compared in Table III. Specifically, Super SloMo is a strong baseline in the multiframe interpolation task. The architecture of the proposed EA-Net is also inspired from it. The main difference of EA-Net and Super SloMo is in the edge-aware processing of frames, including edge-guided flow estimation and edge-protected frame synthesis. The significantly better performance of EA-Net than Super SloMo has demonstrated the necessity of taking edge information into consideration in the multiframe interpolation task. To better estimate the motion in the video, Flawless SlowMotion separates the interpolation task into two parts, including DeblurNet and InterpNet. They are used to deblur the key frames and generate the intermediate frames, respectively. The better performance of Flawless SlowMotion compared with Super SloMo indicates the importance of frame deblurring to the interpolation task. Actually, the proposed EA-Net performs in a similar manner by integrating the edge information into the interpolation task. Better performance of EA-Net has verified the superiority of edge information than frame deblurring.

MEMC-Net and DAIN share similar structures. They combine four encoder–decoder networks to jointly capture different frame information, and another one to synthesize the intermediate frames. Although better results are obtained, they require lots of models pre-trained on other large-scale datasets, which increases the training difficulty and restricts their generality. Fortunately, the proposed EA-Net follows a much simple architecture, where only two encoder–decoder networks are used, and no extra pre-trained models or annotated data are required. Moreover, EA-Net surpasses the performance of MEMC-Net and DAIN. The results have shown that EA-Net is a simple but effective approach in the multiframe interpolation task.

2) Ablation Studies: Table IV shows the results of ablation studies on the Adobe240-fps dataset. Similar to the single-frame interpolation task, three components of EA-Net are analyzed, including the edge-aware mechanisms, attention module, and discriminators. Furthermore, to verify the effectiveness of the flow refinement module, each baseline is provided two results, i.e., with or without the refinement operation. The ablation study in Table IV has verified the contributions of the proposed edge-aware mechanisms. In this case, EA-Net can better preserve object boundaries, which also illustrates the effectiveness of the proposed edge-aware mechanisms. In this case, EA-Net can better preserve object boundaries.

Table IV shows the results of ablation studies on the Adobe240-fps datasets. It can be observed that the edge-aware mechanisms can significantly promote the performance, and the edge augmentation operation is slightly better than the other two, which is similar to the results in the single-frame interpolation task. From the displayed flow maps in Fig. 5, we can see that object boundaries are very clear in the estimated flow maps. It is benefited from our edge augmentation operation. Besides, the variance of the bidirectional flow maps according to time is consistent with the frame motion. It indicates the effectiveness of the flow refinement module, since it can jointly refine and interpolate the flow map.
In Table IV, the attention module can also promote the multiframe interpolation performance. Actually, it can automatically regulate the weight of the forward and backward flow maps in synthesizing the intermediate frames. From the attention maps as depicted in Fig. 5, we can see that the attention weights vary according to time. On one hand, the attention weights are negatively correlated with the intervals between the intermediate frame and the input frames. Concisely, when the intermediate frame is near frame 0, the forward flow map is emphasized, and attention weights of the backward flow map are lower, and vice versa. It is because the shorter interval between two frames indicates smaller frame motions, and the accuracy of flow maps is more likely to be higher. In this case, the attention module can further reduce the interference caused by the low-quality flow maps. On the other hand, the attention maps are clear at object boundaries, so that the edges in synthesized frames are better preserved. Besides, in Table IV, the results of EA-Net with both the frame and edge discriminators are better than the other baselines. It is because the discriminators can promote the reality and clarity of synthesized frames, so that the artifacts and blur are reduced.

Furthermore, in Table IV, the results of EA-Net with and without the refinement module are compared. It can be seen that the refinement module improves the interpolation results significantly, which has verified its necessity. Besides, by comparing the results in Tables III and IV, we can see that EA-Net without the refinement module performs comparably with MEMC-Net and DAIN. Actually, EA-Net without refinement module just contains one encoder–decoder network. It is much more compact than MEMC-Net and DAIN that contain five encoder–decoder networks. The results also demonstrate the superiority of EA-Net in the multiframe interpolation task.

D. Failure Cases and Limitations

Fig. 6 presents several failure cases of EA-Net. In the left two frames, they are already blurred in the ground truth due to the large motion. In the last frame, the rectangular area shares similar texture with the background. In this case, the rectangular areas of the displayed three frames lack edge information, so the blur and artifacts occur. Practically, EA-Net is more suitable for those frames, the displayed three frames lack edge information, so the blur and artifacts are in the red boxes.

REFERENCES

[1] B. Zhao, X. Li, and X. Lu, “Property-constrained dual learning for video summarization,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 10, pp. 5989–5990, Oct. 2020.
[2] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, “Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 6, pp. 3300–3315, Jun. 2022.
[3] B. Zhao, H. Li, X. Lu, and X. Li, “Reconstructive sequence-graph network for video summarization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 5, pp. 2793–2801, May 2022.
[4] X. Shu, L. Zhang, Y. Sun, and J. Tang, “Host–parasite: Graph LSTM-in-LSTM for group activity recognition,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 2, pp. 663–674, Feb. 2021.
[5] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, “Quadratic video interpolation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–10.
[6] H. Sim, J. Oh, and M. Kim, “XVFI: Xtreme video frame interpolation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 14489–14498.
[7] Z. Tu et al., “A survey of variational and CNN-based optical flow techniques,” Signal Process., Image Commun., vol. 72, pp. 9–24, Mar. 2019.
[8] D. Zhang, Q. Huang, Y. Wang, F. Ye, and Y. Wang, “Survey on block-based motion estimation for frame rate conversion,” in Proc. 11th Int. Conf. Digit. Image Process. (ICDIP), Aug. 2019, Art. no. 111794.
[9] T. Ha, S. Lee, and J. Kim, “Motion compensated frame interpolation by new block-based motion estimation algorithm,” IEEE Trans. Consum. Electron., vol. 50, no. 2, pp. 752–759, May 2004.
[10] A.-M. Huang and T. Q. Nguyen, “Multistage motion vector processing method for motion-compensated frame interpolation,” IEEE Trans. Image Process., vol. 17, no. 5, pp. 694–708, May 2008.
[11] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, “AdaCoF: Adaptive collaboration of flows for video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5316–5325.
[12] S. Niklaus and F. Liu, “Softmax splatting for video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5437–5446.
[13] 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, 2020, pp. 1–9.
[14] A. Dosovitskiy et al., “FlowNet: Learning optical flow with convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2758–2766.
[15] D. Sun, 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., Jun. 2018, pp. 8934–8943.
[16] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, “Depth-aware video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3703–3712.
[17] 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.
[18] A.-M. Huang and T. Q. Nguyen, “Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation,” IEEE Trans. Image Process., vol. 18, no. 4, pp. 740–752, Apr. 2009.
[19] L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” ACM Trans. Graph., vol. 23, no. 3, pp. 600–608, Aug. 2004.
[20] C. Dai, Y. Zheng, and X. Li, “Accurate video alignment using phase correlation,” IEEE Signal Process. Lett., vol. 13, no. 12, pp. 737–740, Dec. 2006.
[21] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, “Riesz pyramids for fast phase-based video magnification,” in Proc. IEEE Int. Conf. Comput. Photography (ICCP), May 2014, pp. 1–10.
[22] Y. Yang et al., “Blind identification of full-field vibration modes from video measurements with phase-based video motion magnification,” Mech. Syst. Signal Process., vol. 85, pp. 567–590, Feb. 2017.

[23] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung, “Phase-based frame interpolation for video,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1410–1418.

[24] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers, “PhaseNet for video frame interpolation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 498–507.

[25] F. Arif, S. Amin, A. Ghafoor, and M. M. Riaz, “Frame interpolation in the video domain,” in Proc. Int. Conf. Image Anal. Process., 2019, pp. 249–259.

[26] F. A. Reda et al., “SDC-Net: Video prediction using spatially-displaced convolution,” in Proc. Int. Conf. Image Anal. Process., 2020, pp. 14004–14013.

[27] Y. Zhang, D. Zhao, X. Ji, R. Wang, and X. Chen, “A spatio-temporal autoregressive frame rate up conversion scheme,” in Proc. IEEE Int. Conf. Image Anal. Process., Sep. 2007, pp. 441–444.

[28] Y. Zhang, D. Zhao, X. Ji, R. Wang, and W. Gao, “A spatio-temporal auto-regressive model for frame rate upconversion,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 9, pp. 1289–1301, Sep. 2009.

[29] S. Niklaus, L. Mai, and F. Liu, “Video frame interpolation via adaptive separable convolution,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 261–270.

[30] S. Niklaus, L. Mai, and F. Liu, “Video frame interpolation via adaptive convolution,” in Proc. IEEE Int. Conf. Comput. Vis., Pattern Recognit. (CVPR), Jul. 2017, pp. 670–679.

[31] Y. Liu, Y. T. Liao, Y. Y. Lin, and Y. Y. Chuang, “Deep video frame interpolation using cyclic frame generation,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 8794–8802.

[32] H. Li, Y. Yuan, and Q. Wang, “Video frame interpolation via residue refinement,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 2613–2617.

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

[34] 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. (CVPR), Jun. 2020, pp. 14004–14013.

[35] L. L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, “Motion compensated frame interpolation with a symmetric optical flow constraint,” in Proc. Int. Symp. Vis. Comput., 2012, pp. 447–457.