IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 10, 2020, accepted March 11, 2020, date of publication March 19, 2020, date of current version April 6, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2982039*

# A Stacked Deep MEMC Network for Frame Rate Up Conversion and Its Application to HEVC

**NGUYEN VAN THANG[1], KYUJOONG LEE[2], AND HYUK-JAE LEE[1]**
[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
[2]Department of Electronic Engineering, Sun Moon University, Asan 31460, South Korea

Corresponding author: Kyujoong Lee (kyujoonglee@sunmoon.ac.kr)

This work was supported by the Sun Moon University Research Grant of 2017.

**ABSTRACT** Optical flows and video frame interpolation are considered as a chicken-egg problem such that one problem affects the other and vice versa. This paper presents a stack of deep networks to estimate intermediate optical flows from the very first intermediate synthesized frame and later generate the very end interpolated frame by combining the very first one and two learned intermediate optical flows based warped frames. The primary benefit is that it glues two problems into a single comprehensive framework that learns altogether by using both an analysis-by-synthesis technique for optical flow estimation and Convolutional Neural Networks (CNN) kernels-based frame synthesis. The proposed network is the first attempt to merge two previous branches of previous approaches, optical flow-based synthesis and CNN kernels-based synthesis into a comprehensive network. Experiments are carried out with various challenging datasets, all showing that the proposed network outperforms the state-of-the-art methods with significant margins for video frame interpolation and the estimated optical flows are more accurate for challenging movements. Furthermore, the proposed Motion Estimation Motion Compensation (MEMC) network shows its outstanding enhancement of the quality of compressed videos.

**INDEX TERMS** Frame rate up conversion, video frame interpolation, optical flow, HEVC, MEMC, CNN, convolutional neural networks.

## I. INTRODUCTION

Video frame interpolation is widely used in various applications from computer vision to visual display applications such as frame rate up conversion (FRUC), slow motion display and animation. In order to increase the video frame rate, intermediate frames are generated from two consecutive original frames. Typically, a video frame interpolation algorithm is composed of two distinct steps such that the first step is a motion estimation (ME) [37]–[40] or optical flow (OF) estimation that derives the motion trajectories between two consecutive frames. The second step is motion compensated frame interpolation (MCFI) that synthesizes the intermediate frames by using estimated motion trajectories. The image quality of an interpolated frame depends on the accuracy of the motion trajectories and the performance of the MCFI algorithm. A Block Matching Algorithm (BMA) is widely used for FRUC in Liquid Crystal Displays (LCDs) [33]–[36]. In [34] a hybrid adaptive non-selective block-based MEMC approach is proposed for general cases in MCFI meanwhile a specific semi-global ME method for repetition-like patterns [35] and a hierarchical ME algorithm for small objects [36] are applied independently for challenging cases in MCFI. However, the motion vectors estimated by a BMA are not always the true trajectories of objects because the objective function of BMA is to minimize matching error but does not cover the motion constraints of objects. Consequently, these block-based methods inevitably generate ghost, blocking and blurry artifacts, owing to the errors in estimated motion vectors.

Recently, the break-through of Convolutional Neural Networks (CNN) in computer vision [9]–[12], [14], [15] allows a formulation of video frame interpolation as an end-to-end learning process without optical flow estimation. In those methods, however, the objective function or loss function focuses on only pixel differences. Consequently, it usually fails in the synthesis at areas with fast and/or complex movements which require a critical role of motion estimation for high-quality frame interpolation. Phase-based frame interpolations in [20] and [21] are another approaches to

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan.

generate the intermediate frames without estimating optical flow. However, similar to the above CNN based methods, the phase-based approaches also fail in correct estimation of fast movement.

This paper presents a comprehensive framework that glues two of the above previous approaches into a single stacked network such that an analysis-by-synthesis technique is used to estimate bidirectional intermediate optical flows and later a synthesis network glues intermediate results generated by component branches (an optical flow-based branch and a CNN kernel-based synthesis branch) to synthesize the very end intermediate frame. The primary contributions of the proposed method are summarized as follows.

Firstly, the proposed network is a combination of two branches of approaches: optical flow-based frame interpolation and CNN kernels-based frame synthesis. Secondly, the paper introduces a method to derive directly *Intermediate optical flows* that are the flows from the intermediate frame to two original frames. This module contributes to learning processes for both frame synthesis networks. It glues motionness represented by a reconstruction loss into the pixel matching loss for the first CNN kernels-based synthesis network and it drives the second synthesis network with estimated optical flows. Thirdly, the proposed network is a back-to-back stack of two network layers such that the first network layer generates three input components for the second network layer that is an extended version of SepConv network [15]. Lastly, the proposed method outperforms the previous algorithms for various datasets. Figure 1 shows two examples where the visual quality of the interpolated frames generated by the proposed method is better than that of others obtained by the previous methods, especially in regions with fast, complex movements such as balls. In addition, the proposed network is also applicable to quality enhancement of compressed videos. The proposed method is competitive with the state-of-art network trained specifically for video quality enhancement.

The rest of the paper is organized as follows. Section II reviews previous related works. The proposed network is presented in Section IV. Section V concludes this paper.

## II. PREVIOUS WORKS

### A. VIDEO FRAME INTERPOLATION

Extensive research efforts have been made to handle the challenges in video frame interpolation. A typical approach in video frame interpolation estimates dense motion vector fields, or optical flows, between two original input frames and then interpolates intermediate frames guided by the estimated motions [7], [8], [13], [19], [22]. To synthesize an output image from the input frames, the estimated flows based warping operations using bilinear interpolation are done first, and later the warped frames are blended together. Consequently, the flow-based methods generate ghost or blurry artifacts when the warped frames are not aligned well, owing to the errors of the estimated optical flows. In order to replace simple blending operations, Nikaus and Liu [16] propose to use a context-based synthesis network to generate the intermediate frames from the pre-warped frames. It is shown that the frame synthesis network outperforms simple blending algorithms.

Recently, inspired by the success of applying deep learning to optical flow estimation [6], [27], [29], [30], [32], CNNs are used for video frame interpolation [42]–[45] with the objective function minimizing the pixel difference between the synthesized one and its corresponding ground-truth. CNN-based methods remove optical flow step and handle video frame interpolation as a convolution process [12]–[15], [17], [24]. In other words, the network can be trained to synthesize images without explicit motion estimation step. Consequently, it usually fails at regions with fast and complex moving objects where knowing motion information is crucial for synthesis task. Starting from the work by Long *et al.* in [12] which employs an auto-encoder) network, a number of recently-proposed deep networks successfully improves the quality of frame interpolation. The auto-encoder architecture or U-net architecture used in [15] and [17] extract features that are given to the sub-nets for the synthesis of the intermediate frame. SepConv network in [15] successfully handles blurry artifacts thanks to estimate independently four 1D kernels which are then convolved with the input frames to generate interpolated frames. However, SepConv network does not consider the motion constraints among neighboring kernels because the kernels for each pixel are trained independently from those of neighboring pixels. A deep neural network is also used to directly estimate the phase decomposition of the intermediate frame in [21] based on the application of the phase-based frame interpolation which is originally proposed by Meyer *et al.* in [20] to generate intermediate frames by modifying a per-pixel phase.

### B. A STACK OF NETWORKS

A stack of component networks is proved to improve the performance of the whole network in various tasks including pose estimation [5], object detection [1], document image unwarping [2], optical flow [29] and so on. In [5], stacked hourglass networks are proposed for human pose estimation and they outperform long single hourglass networks as claimed by authors. In [1], the stack of two hourglass networks roles as the backbone network of CornerNet in order to generate features for two prediction modules. In [2], a stacked U-Net with intermediate supervision is used to directly predict the forward mapping between the warped images and the refined version. For optical flow, Flownet 2.0 [29] also employs a stack of several sub-networks and achieves an significant improvement from the previous version. This paper adopts the idea of a stack of sub-networks into video frame interpolation. The proposed stacked network is not only a simple stack of sub-components but it also narrows down the distance between the input frames before feeding to the second sub-network in the stack. In addition, the component sub-networks are not exactly same but they still achieve the


**FIGURE 1.** Visual comparisons of the interpolated frames by previous optical flow based methods, CNN-based interpolation, and the proposed method. The proposed approach achieves the better quality in fast and complex motion of a small object.

same effect of stacked networks because they are very similar in the form of auto-encoder architecture.

### C. CNN KERNEL-BASED SYNTHESIS

SepConv network architecture is shown in Figure 2. The key idea of SepConv and its predecessor, Adaptive [14] is to consider an interpolation process as a convolutional operation. The input frames are convoluted with learned kernels that are final layers in the synthesis network, denoted as, Ker1v, Ker1h, Ker2v, Ker2h in Figure 2 to generate the intermediate frame as shown in equation (1).

$$I_{1.5}(x,y) = Ker_1^v(x,y) * Ker_1^h(x,y) * P_1(x,y) + Ker_2^v(x,y) * Ker_2^h(x,y) * P_2(x,y) \quad (1)$$

where $I_{1.5}(x,y)$ is the intermediate frame to be synthesized, $P_1(x,y)$, $P_2(x,y)$ respectively are the patches centered at $(x,y)$ position in first frame, and second frame. $Ker_1^v(x,y)$, $Ker_1^h(x,y)$, $Ker_2^v(x,y)$, and $Ker_2^h(x,y)$ are the learned pixel-dependent 1D kernels as shown in Figure 2. The advantage of SepConv in comparison with a conventional auto-encoder with direct synthesis [12] as claimed by authors is SepConv alleviates blurry artifacts efficiently and interpolated frames generated by SepConv is sharper than those obtained the symmetric auto-encoder. This paper adopts SepConv as a baseline for the proposed method.

## III. THE PROPOSED NETWORK FOR VIDEO FRAME INTERPOLATION

### A. A STACK OF SYNTHESIS NETWORKS

Analysis-by-synthesis technique for optical flow estimation and CNN kernels-based frame synthesis are the key components of the proposed network that stacks two synthesis networks together, a back-to-back stack to help each other in learning operations. Consequently, it covers both the spatial property of CNN kernels-based synthesis and the temporal property of optical flow-based synthesis. In addition, it also narrows down the displacement between input frames and the final intermediate frame for accurate synthesis.

The proposed network shown in Figure 3, is a back-to-back stack of two synthesis networks. In the front layer, from two original input frames, denoted as $I_1$ for the first frame and $I_2$ for the second frame, the first synthesis network generates the very first intermediate frame, denoted as $I_{1.5}^3$. In addition, as a by-product of the first synthesis network, four 1D kernels, two corresponding to the vertical and horizontal kernels convoluted with the first input frame, denoted as $Ker_1^v$, $Ker_1^h$, the other two kernels for convolution with the second input frame, denoted as $Ker_2^v$ for vertical direction and $Ker_2^h$ for horizontal direction, encode implicitly the motion information and they are used to derive intermediate optical flows by Motion Derivation module. Then, two original input frames are warped to the intermediate time scale using the estimated intermediate optical flows. Finally, three intermediate interpolated frames, the first warped frame, denoted as $I_{1.5}^1$, the second warped frame, denoted as $I_{1.5}^2$ and the very first intermediate frame, $I_{1.5}^3$ are stacked together to feed into the second synthesis network that is a variant of the first one. In term of architecture, the first synthesis network is the same as Sepconv network in [15] in which two inputs are the original frames and outputs are four 1D kernels for convolution with the original frames to generate the very first intermediate frame. The second synthesis network is an extended version of the first synthesis network with the inputs are three intermediate interpolated frames therefore, six 1D kernels are trained to generate the output pixel of the final intermediate frame as the following equation.

$$I_{1.5}(x,y) = K_1^v(x,y) * K_1^h(x,y) * P_{1.5}^1(x,y) + K_2^v(x,y) * K_2^h(x,y) * P_{1.5}^2(x,y) + K_3^v(x,y) * K_3^h(x,y) * P_{1.5}^3(x,y) \quad (2)$$

where $P_{1.5}^1(x,y)$, $P_{1.5}^2(x,y)$, and $P_{1.5}^3(x,y)$ respectively are the patches centered at $(x,y)$ position in intermediate interpolated frames $I_{1.5}^1$, $I_{1.5}^2$ and $I_{1.5}^3$. $K_1^v(x,y)$, $K_1^h(x,y)$, $K_2^v(x,y)$, $K_2^h(x,y)$, $K_3^v(x,y)$, and $K_3^h(x,y)$ are the learned pixel-dependent 1D kernels of the second synthesis network.

The second synthesis network learns from the closest frames to synthesize the final intermediate frame, and it also embraces both optical flow-based results and a CNN kernels-based synthesized frame. Consequently, it can cover challenging motion scenarios, such as fast and complex movements. In addition, the stack of networks is used to narrow down the distance between input frames to estimate condensed interpolation kernels.

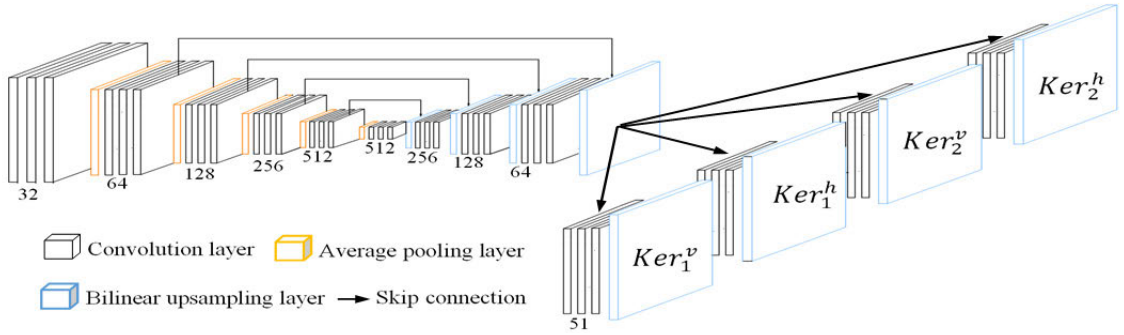



**FIGURE 2.** Architecture of SepConv [15]. Key idea of SepConv is to estimate four 1D pixel-dependent kernels (denoted as $Ker_1^v$, $Ker_1^h$, $Ker_2^v$, and $Ker_2^h$) at final layer that are then convoluted with the input frames to produce the output pixel in a dense pixel-wise manner. The building block of the Convolution layer contains three consecutive convolutional operations followed by a ReLU operation.

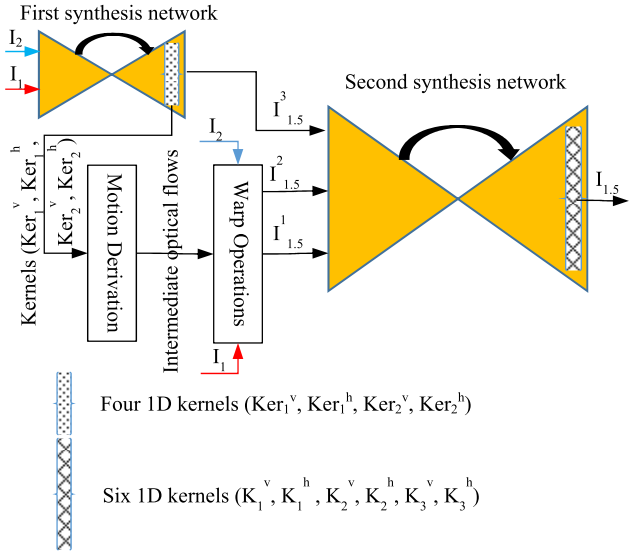



**FIGURE 3.** Architecture of the proposed network. The proposed architecture is a stack of two Sepconv [15] based synthesis networks. In between two synthesis networks are intermediate optical flow estimation module and warp operations in order to glue two networks into a comprehensive frameworks for both tasks, video frame interpolation and optical flow estimation. Consequently, they help each other during training.

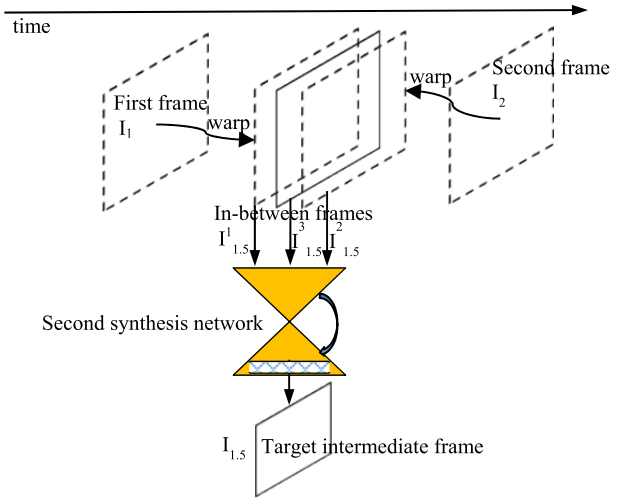



**FIGURE 4.** The time order and structure of the second synthesis network. The first input frame, denoted as $I_1$ and the second input frame, denoted as $I_2$ are warped into intermediate timescale, the warped frames, denoted as $I^1_{1.5}$ and $I^2_{1.5}$ together with the output of network 1, denoted as $I^3_{1.5}$ are fed to the synthesis network 2 with an auto-encoder-based architecture. In term of time order, among in- between frames, $I^3_{1.5}$ is closest to the real output target frame, denoted as $I_{1.5}$, $I^1_{1.5}$ is slightly front offset to $I_{1.5}$, $I^2_{1.5}$ is slightly back offset to $I_{1.5}$. In other words, the triple of intermediate results ($I^1_{1.5}$, $I^3_{1.5}$, $I^2_{1.5}$) is a narrowed down version of the triple of ($I_1$, $I_{1.5}$, $I_2$) with the same time order.

As shown in Figure 4, among three intermediate interpolated frames, in term of time scale, the output of the first synthesis network, denoted as $I^3_{1.5}$ is the nearest to the real output target frame, denoted as $I_{1.5}$. On the other hand, the frame, denoted as $I^1_{1.5}$ that is the warped frame from the first original frame ($I_1$), is a slight offset in the forward direction to the real output target frame, and the frame, denoted as $I^2_{1.5}$ that is the warped frame from the second original frame, ($I_2$), is a slight offset in the backward direction to the real output target frame.

### *B. ANALYSIS BASED SYNTHESIS INTERMEDIATE OPTICAL FLOW ESTIMATIONS*

In the first layer of the stack, the motion derivation module is the glue between two branches of approaches, the optical flow-based frame interpolation and the CNN kernels-based frame synthesis. This solves a chicken-egg problem by training both blended tasks such that the intermediate optical flows, as denoted in Figure 5, are estimated by the analysis-by-synthesis technique through convolution kernels of the first synthesis network. Meanwhile the estimated optical flows role as motion-ness in the loss function of the first synthesis network makes the network learn only pixel matching also motion constraints and scenarios. In addition, estimating the optical flows from the synthesized intermediate frame is a target-based estimation that can fix estimation errors from the previous methods [18], [27] when the intermediate frame is unavailable to verify the accuracy of analysis. In other direction, the estimated intermediate flows are derived from 1D kernels of the first synthesis network. Consequently,

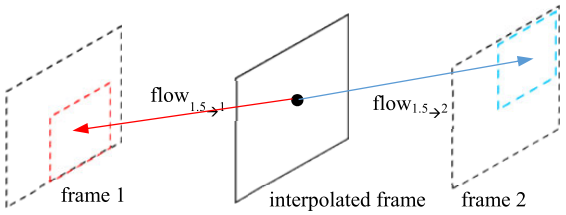



**FIGURE 5.** Bi-directional intermediate optical flows. Flows from the intermediate frames to two original frames called as intermediate flows.

it glues the motion constraints into the first network. Therefore, the first network learns not only pixel matching but also motion information.

The coefficients of 1D kernels implicate motion information and they are exploited to derive the flow information. The motions are encoded as the offsets of the non-zero kernel values to the kernel center. The motion vector is the weighted sum of the offsets. Therefore, the values of the coefficients and the offsets are used in order to compute the motions. There are four 1D kernels, two corresponding to the displacement of First frame, $I_1$, to the interpolated frame, and the others corresponding to the displacement of Second frame, $I_2$, to the interpolated frame. The optical flows for both the forward and backward directions with a point of view from the intermediate frame are computed directly. The formulations of the motion derivation module are represented by the set of equations (3), (4), (5) and (6).

$$u_{1.5\to1} = \frac{\sum_{i=1}^{N} weight_i^{h_1} * offset_i^{h_1}}{\sum_{i=1}^{N} weight_i^{h_1}} \tag{3}$$

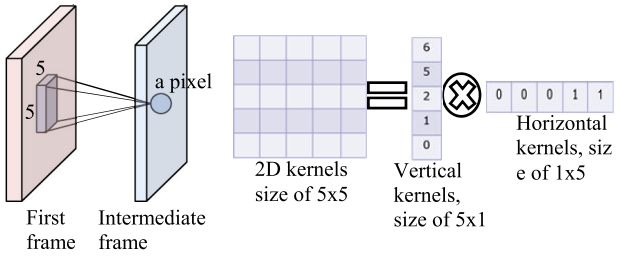
$$v_{1.5\to1} = \frac{\sum_{i=1}^{N} weight_i^{v_1} * offset_i^{v_1}}{\sum_{i=1}^{N} weight_i^{v_1}} \tag{4}$$

$$u_{1.5\to2} = \frac{\sum_{i=1}^{N} weight_i^{h_2} * offset_i^{h_2}}{\sum_{i=1}^{N} weight_i^{h_2}} \tag{5}$$

$$v_{1.5\to2} = \frac{\sum_{i=1}^{N} weight_i^{v} * offset_i^{v_2}}{\sum_{i=1}^{N} weight_i^{v_2}} \tag{6}$$

where $u_{1.5\to1}$ and $v_{1.5\to1}$ are the horizontal and vertical components of the flow from the intermediate frame to First frame, $u_{1.5\to2}$ and $v_{1.5\to2}$ are the horizontal and vertical components of the flow from the intermediate frame to Second frame. $Offset_i{}^h{}_1$, $offset_i{}^v{}_1$, $offset_i{}^h{}_2$, $offset_i{}^v{}_2$ are the displacements of the coefficients to the center position in the corresponding 1D kernels. N is a kernel size and the offset value stands for the motion and the weighted average of offset values is the estimated motion vector.

In order to illustrate for equation (4) (similar for others), let see a toy example in Figure 6. In this example, in order to synthesize a pixel in the intermediate frame, denoted as a tiny circle, an image patch with size of 5 × 5 in the first frame is convoluted with a learned 2D kernel size with size of 5 × 5 that is decomposed into two 1D kernels with size of 5 × 1 for vertical kernel and 1 × 5 for horizontal kernel,




$$v_{1.5\to1} = \frac{\sum_{i=1}^{5} weight_i^{v_1} * offset_i^{v_1}}{\sum_{i=1}^{5} weight_i^{v_1}} = \frac{16}{14} = 1.143$$

where offset vector is $\begin{pmatrix} 2 \\ 1 \\ 0 \\ -1 \\ -2 \end{pmatrix}$ and weight vector is $\begin{pmatrix} 6 \\ 5 \\ 2 \\ 1 \\ 0 \end{pmatrix}$

**FIGURE 6.** A toy example of the derivation of intermediate optical flows.

which means the motion vector is limited from −2 to 2 in both directions. The elements of an offset vector are calculated as the displacements between the position of the weight coefficients and the center position of the vector, as shown in Figure 6. Finally, the motion vector (in this toy example, only vertical direction) is computed as the weighted sum of offset elements where the weights are learned during training process.

### C. WARPING OPERATIONS

Guided by the estimated intermediate optical flows, the proposed method warps the input frames into the intermediate timescale. Backward warping operations, which can be implemented using bilinear interpolation are differentiable. Specifically, the proposed method employs backward warping that uses the estimated backward intermediate optical flow, as denoted as $flow_{1.5\to1}$ in Figure 5, to warp the first input frame, denoted as $I_1$ to the target locations in the intermediate frame and obtains a warped frame, denoted as $I_{1.5}^1$. Likewise, the proposed method warps the second input frame, denoted as $I_2$ and generates the other warped frame, denoted as $I_{1.5}^2$ by using the estimated forward intermediate optical flow, denoted as $flow_{1.5\to2}$ in Figure 5. Two warped frames are very close to the true interpolated frame. Therefore, they are very suitable for the inputs of the second synthesis network that works as a frame refinement to generate the final intermediate frame. This step narrows down the distances between two consecutive input frames and the intermediate one. In addition, it is easier for the network to learn kernels when two inputs are closer.

### D. LOSS FUNCTIONS AND TRAINING

The proposed network is a stack of component subnets, as suggested by [5], [29], in order to avoid over-fitting, the proposed network should be trained end-to-end with a loss function that contains a *Final loss*, denoted as $||I_{1.5} - I_{gt}||_1$ in equation (7) and an *Intermediate loss*, denoted as $||I_{1.5}^3 - I_{gt}||_1$ in equation (8), where $I_{gt}$ is the ground truth intermediate

**TABLE 1. Objective comparisons on middebury benchmark.**

| | Mequon | Schefflera | Urban | Teddy | Backyard | Basketball | Dumptruck | Evergreen | Average |
|---|---|---|---|---|---|---|---|---|---|
| Proposed | 2.61 | 3.30 | 3.14 | 4.74 | **8.11** | **4.48** | **5.78** | **6.06** | **4.78** |
| CtxSyn [16] | **2.24** | **2.96** | 4.32 | **4.21** | 9.59 | 5.22 | 7.02 | 6.66 | 5.28 |
| MDP-Flow2 [19] | 2.89 | 3.47 | 3.66 | 5.20 | 10.2 | 6.13 | 7.36 | 7.75 | 5.83 |
| SuperSlomo [18] | 2.51 | 3.66 | **2.91** | 5.05 | 9.56 | 5.37 | 6.69 | 6.73 | 5.31 |
| SepConv [15] | 2.52 | 3.56 | 4.17 | 5.41 | 10.2 | 5.47 | 6.88 | 6.63 | 5.61 |
| DeepFlow [22] | 2.98 | 3.88 | 3.62 | 5.39 | 11.0 | 5.91 | 7.14 | 7.80 | 5.97 |

frame. The loss function represented in equation (9) is a sum of the final loss and the intermediate loss, it is called as *Pixel matching loss*. However, as its name, this function only contains the pixel matching losses for synthesis networks, there is no component to represent for the optical flow loss. To glue the optical flow loss into the total loss function, we propose to add a *Reconstruction loss* that represents for optical flow components in the proposed comprehensive network into the total loss. This reconstruction loss is computed as equation (10)

$$\textit{Final loss} = ||\mathrm{I}_{1.5} - \mathrm{I}_{gt}||_1 \tag{7}$$

$$\textit{Intermediate loss} = ||\mathrm{I}^3_{1.5} - \mathrm{I}_{gt}||_1 \tag{8}$$

$$\textit{Pixel matching loss} = ||\mathrm{I}_{1.5} - \mathrm{I}_{gt}||_1 + ||\mathrm{I}^3_{1.5} - \mathrm{I}_{gt}||_1 \tag{9}$$

$$\textit{Reconstruction loss} = ||\mathrm{I}^w_{1.5} - \mathrm{I}_{gt}||_1 \tag{10}$$

where $\mathrm{I}^w_{1.5} = (\mathrm{I}^1_{1.5} + \mathrm{I}^2_{1.5})/\,2.0$ represents for both the warped intermediate frames, $\mathrm{I}^1_{1.5}$ and $\mathrm{I}^2_{1.5}$ obtained by warping operations. Consequently, the total loss of the proposed network is trained as equation (11)

$$\begin{aligned}\textit{Total loss} &= \textit{pixel matching loss} + \textit{Reconstruction loss} \\ &= ||\mathrm{I}_{1.5} - \mathrm{I}_{gt}||_1 + ||\mathrm{I}^3_{1.5} - \mathrm{I}_{gt}||_1 + ||\mathrm{I}^w_{1.5} - \mathrm{I}_{gt}||_1\end{aligned} \tag{11}$$

Following [15], [25] the proposed neural network parameters are initialized by a convolution aware initialization [31] and trained by using AdaMax [26] with $\beta 1 = 0.9$, $\beta 2 = 0.999$, a learning rate of 0.001 and a mini-batch size of 12 samples. Because the motion derivation module derives the intermediate optical flows from four 1D learned kernels from the first synthesis network, if the kernels are estimated wrongly, it results in the wrong motion vectors. Consequently, the learning process takes more time to converge. Therefore, in order to speed up the training process, in the very first epochs of the training, the total loss function is assigned equally to the loss function of the first synthesis network. In other words, the total loss function in the very first epochs is equal to *Intermediate loss*, $||\mathrm{I}^3_{1.5} - \mathrm{I}_{gt}||_1$. After five epochs, the whole proposed network is trained with the total loss described by equation (11). The training dataset provided by [23] is used to train the proposed network because this dataset contains high-quality frames extracted from high-resolution videos downloaded from vimeo.com. The resolution of training videos is 448 × 256. For data augmentation during the training process, the trainer randomly swaps the temporal order between input frames, First frame becomes Second frame and vice versa. This makes dataset larger and eliminates potential priors. Pytorch library is used to train the proposed network with two NVIDIA GTX 1080 GPUs. For inference stage, in a single graphic card, it takes 0.86 seconds to generate an interpolated frame with the resolution of 1280 × 720 and 0.55 seconds to interpolate an intermediate frame with the resolution of 640 × 480.

**TABLE 2. Objective comparisons on Vimeo90K dataset among CNN-based methods.**

| | PSNR | SSIM |
|---|---|---|
| ToFlow [23] | 33.53 | 0.9668 |
| ToFlow+mask [23] | 33.73 | 0.9682 |
| SepConv [15] | 33.85 | 0.9697 |
| Proposed | **34.65** | **0.9737** |

**TABLE 3. Objective comparison on UCF101 dataset.**

| | PSNR | SSIM |
|---|---|---|
| Frame average | 33.14 | 0.9519 |
| PWC-Net [27] | 33.76 | 0.9618 |
| MDP-Flow2 [19] | 34.52 | 0.9660 |
| DVF [17] | 34.12 | 0.9631 |
| SepConv [15] | 34.78 | 0.9669 |
| Super-Slomo [18] | 34.75 | 0.9669 |
| Proposed | **34.96** | **0.9683** |

## IV. EXPERIMENTAL RESULTS

### A. FRAME INTERPOLATION EVALUATIONS

To evaluate the proposed network, quantitative and qualitative comparisons with several representative state-of-the-art video frame interpolation and optical flow methods are made. Firstly, evaluations are carried out with the interpolation category of Middlebury optical flow benchmark that is typically used for assessing frame interpolation methods [13]. The proposed approach is compared with the methods that rank high with this interpolation benchmark. The first one is MDP-Flow2 [19], an accurate optical flow method, as it still remains the highest rank among all classic optical flow methods with the Middlebury benchmark. In addition, PWC method [27] that is a state-of-the-art CNN-based optical flow algorithm that performs the top among CNN-based methods ranked with well-known Sintel optical flow benchmark [3]. To synthesize interpolated frames from the computed optical flows, the same algorithm in [13] is used. For a CNN-based frame synthesis algorithm without optical flow estimation,

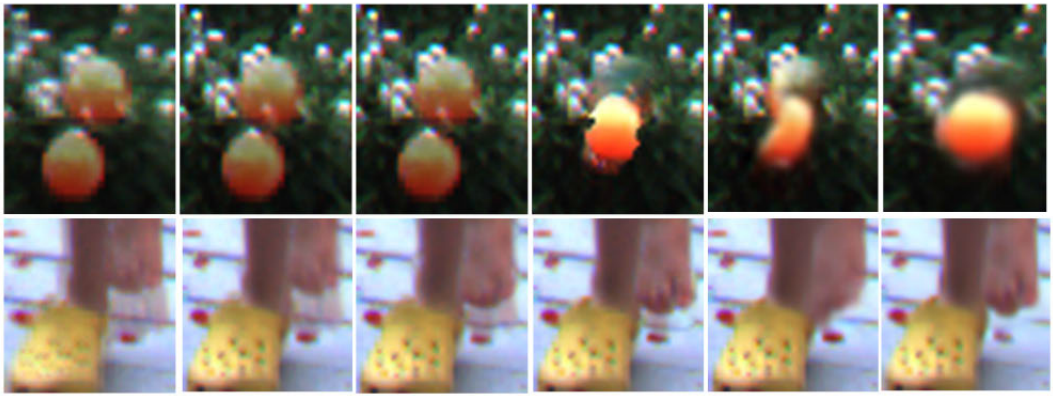



**FIGURE 7.** Visual comparisons on Backyard sequence on Middlebury benchmark.

**TABLE 4.** Objective comparison on HCD dataset.

| | MDP-Flow2 [19] | | PWC-Net [27] | | SepConv [15] | | Proposed | |
|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Subtitle | 31.83 | 0.9913 | 24.82 | 0.9661 | 33.83 | 0.9924 | **34.73** | **0.9929** |
| Occlusion | 29.81 | 0.9555 | 28.35 | 0.9476 | 30.92 | 0.9622 | **32.29** | **0.9706** |
| Soccer | 29.38 | 0.9599 | 28.01 | 0.9479 | 29.79 | 0.9636 | **31.04** | **0.9702** |
| Basketball | 34.36 | 0.9867 | 31.11 | 0.9720 | 34.84 | 0.9876 | **36.14** | **0.9902** |
| Average | 31.35 | 0.9734 | 28.07 | 0.9584 | 32.35 | 0.9765 | **33.55** | **0.9810** |

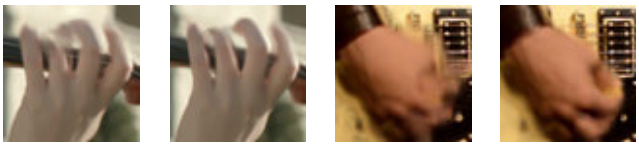



**FIGURE 8.** Visual comparison between SepConv and the first network.

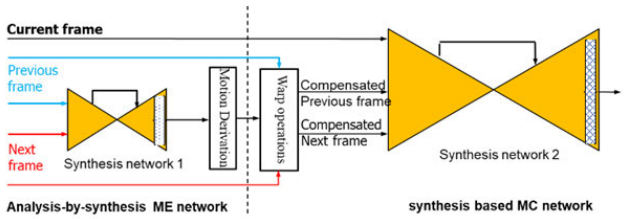



**FIGURE 9.** Architecture of our MEMC network.

recent SepConv [15] method is chosen owing to its high performance among CNN-based algorithms. The optical flow that is a by-product of the proposed network is also compared to state-of-the-art methods.

Table 1 shows the average interpolation error (AIE) used in [13] where the interpolation error is the root-mean-square (RMS) difference between the ground-truth image and the estimated interpolated image. The proposed network outperforms state-of-art methods and improves the best previous method by a significant margin (9.5%) in term of average AIE among eight test images. Especially with *Backyard*, *Basketball, Dumptruck* and *Evergreen* datasets which show real-world scenes, captured with a real camera and containing real sources of noise, the proposed network achieves the best result consistently by notable margins. The proposed interpolation method, denoted as FRUCnet, is ranked the 3$^{rd}$ in Interpolation Error among over 150 algorithms in the benchmark website at the submission time. For visual evaluation, Figure 7 shows the proposed interpolated frame that generates a clear image alleviating ghost and distorted artifacts whereas the previous algorithms still have those artifacts in the interpolated frames.

The next well-known dataset for evaluating video frame interpolation algorithms is Vimeo90K dataset provided by [23]. It contains 3,782 triplets of frames with the image resolution of 448 × 256 pixels. As shown in Table 2, the proposed method outperforms previous CNN-based networks, SepConv, ToFlow and its variant, ToFlow with a mask by significant margins in terms of both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [4].

UCF101 dataset [28] consists of videos with the size of 256 × 256. This dataset is initially used to evaluate activity recognition and later it is used to evaluate the frame interpolation originated from [17]. UCF101 dataset includes videos with small motion. Therefore, even with a simple interpolation algorithm such as frame average, the video quality of an interpolated frame is sufficiently high as shown in Table 3. In this dataset, the proposed network also outperforms other previous methods. Visual comparison on UCF101 results are shown in Figure 12.

The last one is a new dataset proposed in this paper to cover the difficult cases for frame interpolation. These cases include the movement of text objects, occlusion, reveal, and complex movements of small and fast-moving objects. Movement of text objects as a subtitle and logos is difficult for interpolation because the movement often takes place in a background while its motion is in a different direction from the background. Object occlusion and reveal are difficult in

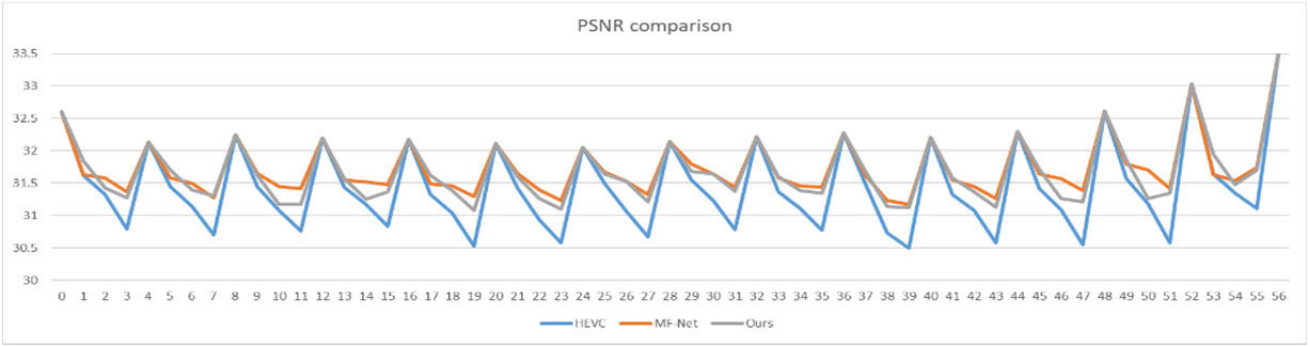



**FIGURE 10.** An example of frame by frame comparison.

**TABLE 5.** Effect of the reconstruction loss.

| Dataset | PSNR (dB) | | |
|---|---|---|---|
| | Without *reconstruction loss* and *intermediate loss* | Without *reconstruction loss* | Fully total loss |
| UCF101 | 33.03 | 34.76 | **34.96** |
| Vimeo90K | 33.21 | 34.55 | **34.65** |
| HCD | 31.68 | 33.49 | **33.55** |

**TABLE 6.** Comparison between SepConv and the first synthesis network.

| | SepConv [15] | | the first synthesis network | |
|---|---|---|---|---|
| Metric | PSNR | SSIM | PSNR | SSIM |
| Subtitle | 33.83 | 0.9924 | **33.93** | **0.9925** |
| Occlusion | 30.92 | 0.9622 | **31.53** | **0.9666** |
| Soccer | 29.79 | 0.9636 | **30.06** | **0.9686** |
| Basketball | 34.84 | 0.9876 | **35.32** | **0.9895** |
| Average | 32.35 | 0.9765 | **32.71** | **0.9793** |


**FIGURE 11.** Visual comparison between reconstructed frames.

a classical computer vision problem such as optical flow and they are also difficult in frame interpolation. A small object is difficult to estimate its motion and so is fast and complex movement. This new dataset is used to measure the performance of frame interpolation algorithms that focus on enhancement of visual quality. For explanation, this new data set is called *Hard Cases for Display (HCD)* which consists of four video sequences, each contains 60 frames with the resolution of 864 × 480, except the Basketball sequence that contains 90 frames with the resolution of 416 × 240. The dataset covers hard and challenging cases for frame interpolations such as scenes with sub-title, occlusion and reveals, fast complex motions, and the movement of small objects. Table 4 shows quantitative comparisons between the proposed methods with representative state-of-the-art methods on HCD dataset. In both PSNR and SSIM, the proposed method outperforms state-of-art methods with notable margins. Figure 13 shows the interpolated frames for visual quality comparisons. In a fast and complex motion sequence, as shown in the top row of Figure 13, the movement of the leg of the soccer player and that of the hand of the goalkeeper is fast and complex. The proposed method improves significantly visual quality in comparison with the previous methods. The middle rows show the interpolated frames for the subtitle sequence where the text objects in the subtitle region include artifacts. The previous methods based on optical flow estimations, and CNN kernel based SepConv, suffer from these artifacts whereas the proposed method successfully removes them. For small objects such as balls in the basketball sequence shown in the bottom row, the proposed network alleviates ghost artifacts significantly when compared with the previous methods.

### *B. PERFORMANCE ANALYSIS*

#### 1) OFTICAL FLOW EVALUATION

Figure 14 shows comparisons between the estimated optical flow by the proposed method and the results obtained by state-of-the-art optical flow methods including MDP-Flow2 [19] (the top-ranked in the Middlebury benchmark) and recent CNN based flow networks, SPyNet [30] and PWC-Net [27]. The top row shows the estimated optical flow results, and the bottom row is the corresponding interpolated frame generated by the above flows by using the same frame interpolation algorithm [13]. The proposed analysis-by-synthesis based motion derivation module estimates the movement of

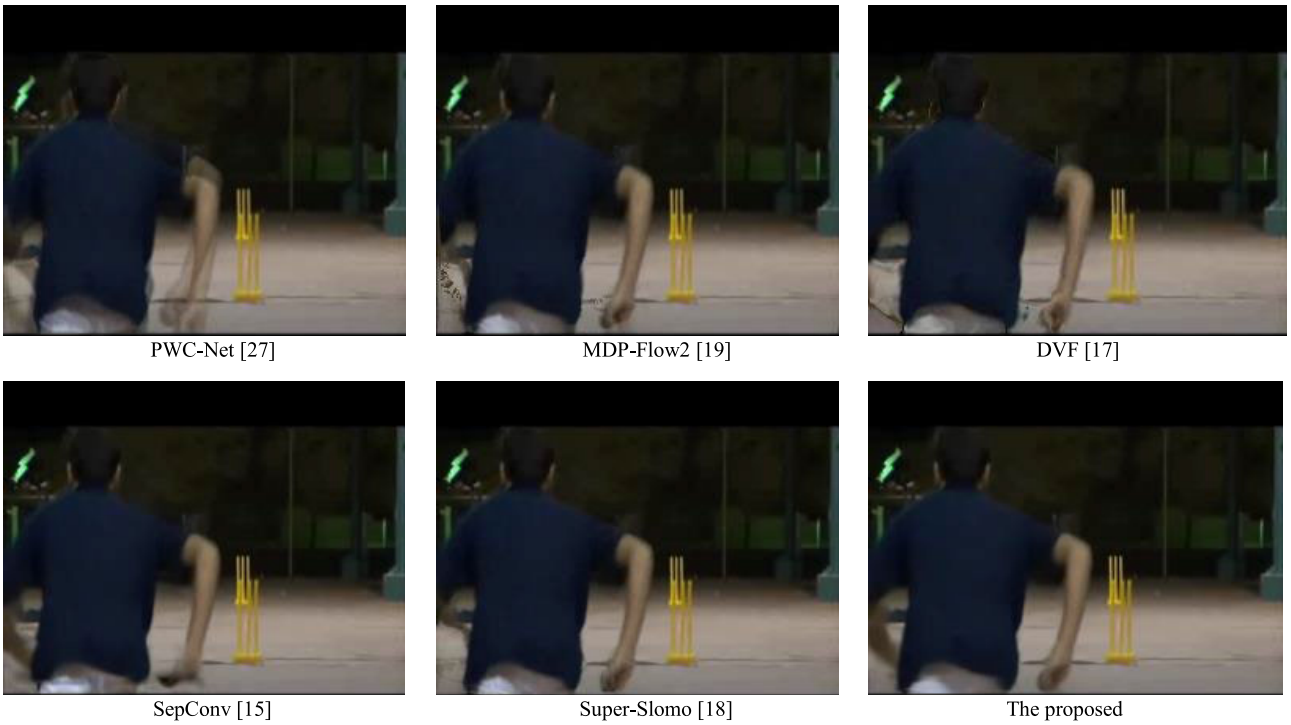



**FIGURE 12.** Visual comparison of interpolated frames on the UCF101 dataset.

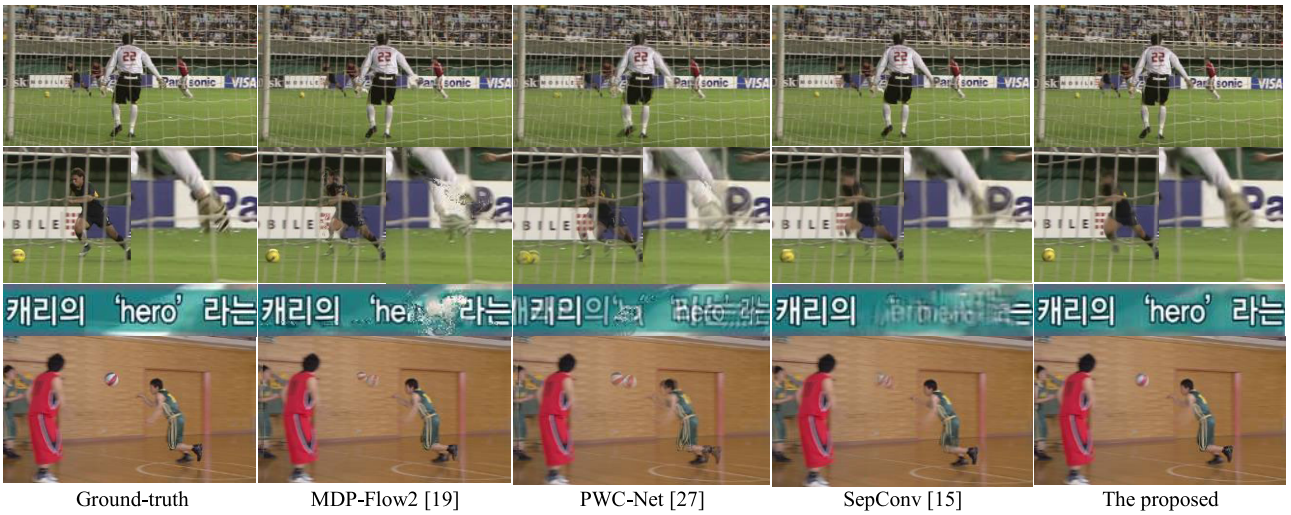



**FIGURE 13.** Visual comparison of interpolated frames on the HCD dataset.

the rotating and moving balls accurately. The results prove that the proposed method preserves the motion of small objects such as balls meanwhile others fail to estimate the movement of the ball. Consequently, either two balls or a distorted ball artifact appears in the interpolated frames generated by the previous methods.

### 2) CONTRIBUTION OF THE LOSS TERMS

Table 5 shows the contribution of each loss term, the reconstruction loss and the intermediate loss on the performance of the whole network. When the network is trained with appearance of the optical flow components, it glues two synthesis networks more coherently for learning. As a result, the quality of the interpolated frame is enhanced by integrating the reconstruction loss into the total loss during training. In addition, when the intermediate loss is removed from the fully total loss function, the performance of the network is reduced significantly, this re-affirms that the network with stack architecture is over-fitting when learning without intermediate loss function.

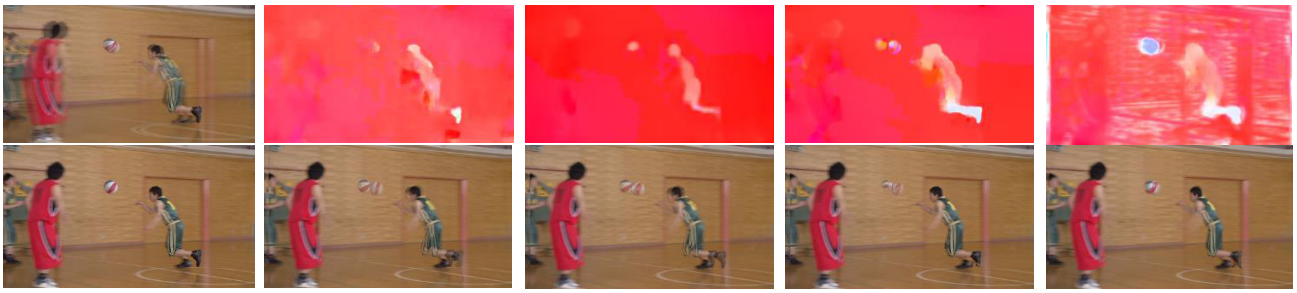



**FIGURE 14.** Visual optical flow results on basketball sequence. Top row is the overlaid frames, color encoded optical flow images of methods. Bottom row is the ground-truth frame and interpolated frames generated by respectively above flows with the same frame interpolation algorithm in [13].

#### 3) COMPARISON BETWEEN THE FIRST SYNTHESIS NETWORK AND SepConv

The effect of the motion-ness on the performance of the first synthesis network is also evaluated. The motion guided warping operations and the first synthesis network are glued together by motion derivation module that adds a motion-ness into the pixel matching loss, this makes the first synthesis network learn not only pixel matching but also motion constrains and scenarios, meanwhile the Sepconv network [15] is similar to a pixel or patch matching that only learns for a pixel loss. Table 6 and Figure 8 show that the first synthesis network outperforms SepConv in both objective comparisons and subjective visual evaluations.

### C. EXTENSION FOR THE QUALITY ENHANCEMENT OF COMPRESSED VIDEO TASK

Interestingly, the proposed video frame interpolation network is designed for video frame interpolation or frame rate up conversion problem, but it can apply for post-processing task of compressed videos in order to improve quality of the compressed videos. One more time, it shows the generalization of the proposed network. In this section, the proposed video frame interpolation network described Section III is called as a MEMC network that composes of two synthesis networks, motion derivation and warp operations modules. The input of the proposed MEMC network are three reconstructed frames, the current reconstructed frame that to be enhanced and two nearest neighbor reconstructed frames, one is the previous frame, the other is the next frame. The first synthesis network and motion derivation module role as a motion estimation network that will derive motion vectors between the current frame and two nearest neighbor frames. On the other hand, the warp operations and the second synthesis network role as a motion compensation network that generates the enhanced current frame from three inputs, the current frame, the motion compensated previous frame, and the motion compensated next frame. As described in Section III, the proposed motion estimation network is an analysis-by-synthesis technique that estimates motion more accurate than conventional analysis based approaches. In addition, unlike MF-Net [41] for this task, that only work well for low delay configuration, the proposed MEMC network can be applied for both low delay configuration and random access configuration. The proposed MEMC network, shown in Figure 9 is compared to the previous state-of-art MF-Net [41] that is designed specifically for this task. Video sequences are compressed by the same HEVC reference software (HM 16.0) with Quantization Parameter (QP) = 37. Table 7 shows comparison on the quality of reconstructed sequence, among the baseline HEVC, the enhanced sequence by the previous MF-Net and the enhanced sequence obtained by the proposed MEMC network. Both MF-Net and the proposed network improve quality of reconstructed frames significantly, 0.41 dB and 0.35 dB respectively.

**TABLE 7.** Comparison on the quality of recontructed frames.

| PSNR (dB) | | |
|---|---|---|
| HEVC | MF-net [41] | Ours |
| 31.09 | **31.50** | 31.44 |

In order to have an insight analysis on how each frame is improved by each method, Figure 10 shows the examples of frame by frame comparison between methods. The blue line denotes the PSNR of reconstructed frames obtained by base line HEVC (or decoded by HEVC), the orange line denotes the PSNR of enhanced frames by applying MF-Net, and the grey line denotes the PSNR of enhanced frames obtained by the proposed MEMC network. The period of each anchor frame is four, that means frames 0, 4, 8, 12, 16 and so on are anchor frames. Lowest quality reconstructed frames are frames just before the corresponding anchor frames, such as frame 3, 7, 11, 15 and enhanced significantly by both post-processing networks, MF-Net and the proposed MEMC network, which show 0.68 and 0.57 dB increase from baseline HEVC, respectively. The proposed method outperforms the MF-Net at frames next right after anchor frames such as frame 1, 5, 9, 13, 17 and so on. Because in MF-Net approach, distance from those frames to corresponding anchor frames are asymmetric. For middle frames between anchor frames, such as frame 2, 6, 10, 14, and so on, the MF-Net improves better than the proposed MEMC network, 0.36 dB improvement,

compare to 0.22 dB improvement obtained by the proposed MEMC network.

Figure 11 shows a visual comparison between enhanced reconstructed frame obtained by MF-Net and that obtained the proposed MEMC network. Both enhanced frames alleviate blocking artifacts significantly in comparison to the raw reconstructed frame decoded by HEVC baseline, especially at regions around the ball.

## V. CONCLUSION

This paper proposes a back-to-back stack of synthesis networks by bridging the gap between two branches, optical flow based synthesis and learned CNN kernels-based interpolation together into a comprehensive joint framework. Intermediate optical flows are introduced and estimated directly from learned CNN kernels by using analysis-by-synthesis technique and the first synthesis network learns not only a pixel matching loss but also motion-ness criterion. Consequently, the proposed method handles fast, complex motions of small objects effectively. The proposed network is also the first attempt to bridge two branches of previous approaches, optical flow based synthesis and CNN kernels based synthesis into a comprehensive network. The proposed method is evaluated with various datasets and outperforms previous methods in both objective metrics and subjective visual evaluations. When it is applied for the quality enhancement of compressed videos, it completes with the state-of-the-art network that is trained specifically for this task.

## REFERENCES

[1] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, Mar. 2020.
[2] K. Ma, Z. Shu, X. Bai, J. Wang, and D. Samaras, "DocUNet: Document image unwarping via a stacked U-Net," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4700–4709.
[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 611–625.
[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[5] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 483–499.
[6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
[7] L. L. Raket, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," *Adv. Vis. Comput.*, vol. 7431, pp. 447–457, Jul. 2012.
[8] Z. Yu, H. Li, Z. Wang, Z. Hu, and C. W. Chen, "Multi-level video frame interpolation: Exploiting the interaction among different levels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1235–1248, Jul. 2013.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[12] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9910, Oct. 2016, pp. 434–450.
[13] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, Mar. 2011.
[14] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 670–679.
[15] 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.
[16] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1701–1710.
[17] 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. (ICCV)*, Oct. 2017, pp. 4463–4471.
[18] 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.
[19] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, Sep. 2012.
[20] 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.
[21] 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.
[22] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1385–1392.
[23] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," 2017, *arXiv:1711.09078*. [Online]. Available: http://arxiv.org/abs/1711.09078
[24] T. Xue, J. Wu, K. L. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 91–99.
[25] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan, "PixelNet: Representation of the pixels, by the pixels, and for the pixels," 2017, *arXiv:1702.06506*. [Online]. Available: http://arxiv.org/abs/1702.06506
[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[27] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.
[28] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," in *Proc. CRCV*, 2012, pp. 1–7.
[29] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2462–2470.
[30] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4161–4170.
[31] A. Aghajanyan, "Convolution aware initialization," 2017, *arXiv:1702.06295*. [Online]. Available: https://arxiv.org/abs/1702.06295
[32] Y. Lu, J. Valmadre, H. Wang, J. Kannala, M. Harandi, and P. H. S. Torr, "Devon: Deformable volume network for learning optical flow," 2018, *arXiv:1802.07351*. [Online]. Available: http://arxiv.org/abs/1802.07351
[33] D. Wang, L. Zhang, and A. Vincent, "Motion-compensated frame rate up-Conversion—Part I: Fast multi-frame motion estimation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 133–141, Jun. 2010.
[34] N. Van Thang and H.-J. Lee, "An efficient non-selective adaptive motion compensated frame rate up conversion," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
[35] N. Van Thang and H.-J. Lee, "A semi-global motion estimation of a repetition pattern region for frame interpolation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2563–2566.

[36] N. Van Thang, J. Choi, J.-H. Hong, J.-S. Kim, and H.-J. Lee, "Hierarchical motion estimation for small objects in frame-rate up-conversion," *IEEE Access*, vol. 6, pp. 60353–60360, 2018.
[37] C. Bartels and G. de Haan, "Smoothness constraints in recursive search motion estimation for picture rate conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1310–1319, Oct. 2010.
[38] H. Liu, R. Xiong, D. Zhao, S. Ma, and W. Gao, "Multiple hypotheses Bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1188–1198, Aug. 2012.
[39] D. Choi, W. Song, H. Choi, and T. Kim, "MAP-based motion refinement algorithm for block-based motion-compensated frame interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 10, pp. 1789–1804, Oct. 2016.
[40] T. Tsai and H. Lin, "Hybrid frame rate up conversion method based on motion vector mapping," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 11, pp. 1901–1910, Jun. 2013.
[41] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6664–6673.
[42] 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.
[43] Y.-L. 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, Jul. 2019, pp. 8794–8802.
[44] F. Reda, D. Sun, A. Dundar, M. Shoeybi, G. Liu, K. Shih, A. Tao, J. Kautz, and B. Catanzaro, "Unsupervised video interpolation using cycle consistency," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 892–900.
[45] T. Peleg, P. Szekely, D. Sabo, and O. Sendik, "IM-net for high resolution video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2398–2407.

**NGUYEN VAN THANG** received the B.S. degree in electrical engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2010, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2012 and 2019, respectively. His research interests are in the areas of image/video processing, focus on motion analysis and video frame rate up conversion and deep learning for image/video processing applications.

**KYUJOONG LEE** received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2002, the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2008, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, in 2013. From 2002 to 2005, he was with Com2us Corporation, Seoul, as a Developer. From 2013 to 2017, he worked for S.LSI division of Samsung Electronics Corporation. In 2017, he was appointed as an Assistant Professor with the Department of Electronic Engineering, Sun Moon University, Asan, South Korea. His major research interests include the algorithms and architectures of deep learning and image/video processing.

**HYUK-JAE LEE** received the B.S. and M.S. degrees in electronics engineering from Seoul National University, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1996. From 1998 to 2001, he worked with the Server and Workstation Chipset Division, Intel Corporation, Hillsboro, OR, USA, as a Senior Component Design Engineer. From 1996 to 1998, he was on the faculty of the Department of Computer Science, Louisiana Tech University, Ruston, LA, USA. In 2001, he joined the School of Electrical Engineering and Computer Science, Seoul National University, where he is currently working as a Professor. He is also a Founder of Mamurian Design, Inc., a fabless SoC design house for multimedia applications. His research interests are in the areas of computer architecture and SoC design for multimedia applications.