This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

Weighted Convolutional Motion-Compensated Frame Rate Up-Conversion Using Deep Residual Network

Yongbing Zhang, Lixin Chen, Chenggang Yan, Peiwu Qin, Xiangyang Ji, Qionghai Dai

***Abstract*—Frame rate up-conversion (FRUC) usually suffers unreliable motion vectors due to the absence of the current frame to be interpolated. Besides, since the majority of video sequences are usually compressed by various coding standards to reduce the data volume, the quality of generated frames in FRUC will be further impaired. To address this problem, we proposed two FRUC algorithms based on deep residual network. We first present a deep residual network for FRUC (DRNFRUC), which consists of feature extraction, feature recursive analysis, and image restoration parts with a skip connection between the input and output of the network. The proposed DRNFRUC takes the result of an arbitrary existing FRUC method as the input and is able to significantly reduce the edge blurring and blocking artifacts when the motion of the block is violent. In addition, we proposed a deep residual network with weighted convolutional motion compensation (DRNWCMC) for FRUC, where the convolution operations can be embedded into the motion compensation interpolation (MCI) in any existing MCI based FRUC method. In DRNWCMC, we first devise two convolutional neural networks corresponding to the forward and backward motion compensated frames, respectively. And then the adaptive interpolation coefficients for motion compensation are designed as two 1x1 convolutional kernels. Finally, the interpolation result of WCMC is fed into another convolutional neural network to further improve the performance. All the parameters involved in DRNWCMC are trained simultaneously under the same cost function. Experimental results show that the two proposed algorithms can remarkably improve both objective and subjective quality of interpolated frames.**

***Index Terms*—frame rate up-conversion (FRUC), residual network, convolutional neural network.**

I. Introduction

FRAME rate up conversion (FRUC) refers to a frequently used technique that makes use of two original adjacent frames to insert an interpolated frame between them in order to improve the visual quality by increasing the frame rate [1]. Recently, FRUC is embedded in many applications such as digital television, liquid crystal display (LCD), distributed video coding (DVC) and video conferencing [2], [3], [4].

In the past decades, some simple FRUC algorithms generate interpolated frames by copying the previous frame or averaging the previous frame and the following frame. However, these simple algorithms fail to generate satisfactory interpolated frames since they neglect the motions among successive frames. As the research progresses, motion compensated frame rate up-conversion (MC-FRUC) was proposed to improve the quality of interpolated frames. MC-FRUC which takes object's movement into consideration generates more satisfactory results in existing FRUC algorithms [5]. The most significant steps of MC-FRUC are motion estimation (ME) and motion compensated interpolation (MCI) [6]. ME finds motion vectors for each frame to be interpolated by comparing the similarity of two or more successive frames. MCI generates an intermediate frame with motion vectors obtained in the ME step.

Better motion vectors generated from ME will lead to interpolated frame with higher visual quality. In order to reduce visible artifacts, several strategies are proposed to obtain better motion vectors. The most widely-applied algorithm is the block matching algorithm (BMA) [7], [8], [9], [10] due to its hardware-friendly implementation and intuitive architecture. The algorithms of ME based on BMA can be divided into two categories: unilateral ME (UME) and bilateral ME (BME) [11], [12]. UME usually generates incorrect motion vectors that can not reflect the true motion of objects and results in overlaps and holes. BME mitigates the problems by combining motions of the previous and the following frames. Thus, most of FRUC algorithms employ BME [13], [14], [15], [16] to reduce computational complexity and improve the accuracy of motion vectors. Nevertheless, serious artifacts [17] [18] can be observed when the motion vectors of neighboring blocks are not consistent. To further reduce the blocking artifacts and improve the quality of interpolated frame, many advanced motion compensation methods were proposed, e.g., overlapped block motion compensation (OBMC) [19], [11], [20], [21], the weighted index-based bidirectional MCI (WMCI) [13] and the direction-select ME based OBMC (DS-OBMC) [14].

However, due to the absence of the frame to be interpolated, it is very difficult to find the accurate motion vectors in FRUC. Besides, most videos are usually compressed by various coding standards to reduce the huge data volume of the original

Manuscript received Feb 24, 2018; accepted Nov 29, 2018. This work was supported in part by the Projects of National Science Foundations of China (61571254), Shenzhen Fundamental Research fund (JCYJ20170817161409809), and Guangdong Special Support Plan (2015TQ01X16). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Liu Shan. (*Correspoding author: Chenggang Yan*)

Y. Zhang is with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: zhang.yongbing@sz.tsinghua.edu.cn).

L. Chen and C. Yan are with the Department of Automation, Hangzhou Dianzi University, Hangzhou, 310018, China (e-mail: lixinhdu@gmail.com; cgyan@hdu.edu.cn).

P. Qin is with Tsinghua-Berkeley Shenzhen Institute, 518055, China (e-mail: peiwu@sz.tsinghua.edu.cn).

X. Ji and Q. Dai are with TNLIST and Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: xyji@tsinghua.edu.cn; qhdai@tsinghua.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

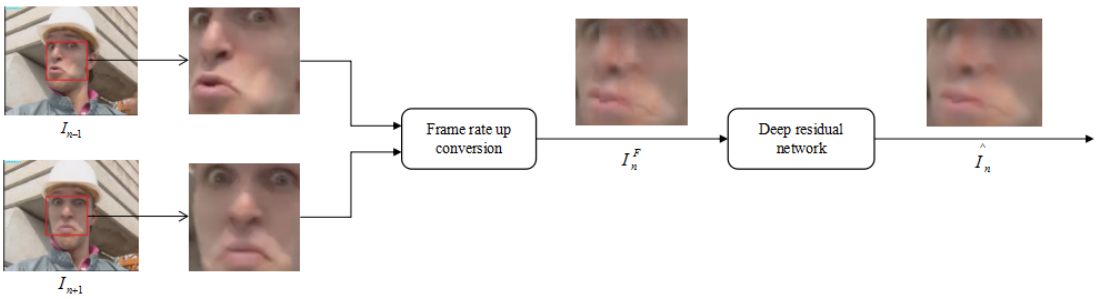


Fig. 1: Frame Rate Up Conversion Using Deep Residual Network

video, which further limits the quality of interpolated frames. To remove artifacts and improve performance, Wang et al. [22] proposed a decoder-end post-processing deep convolution network was investigated to boost the quality of decoded frames. Jia et al. [23] proposed a spatial-temporal residue network based in-loop filter to improve visual quality. Song et al. [24] proposed a practical convolutional neural network as the loop filter for intra frames. However, the existing methods have not been fully exploited in the scenario of bidirectional motion compensation. To address this problem, we proposed a deep residual network with weighted convolutional motion compensation, which can utilize the spatial and temporal information of forward motion compensated frame and backward motion compensated frame. The algorithms are able to improve the performance of interpolated frames over compressed videos and mitigate the ghost and fuzzy effects in edge regions when compared with the traditional FRUC algorithms.

This paper is organized as follows. Section II introduces the related work. Section III describes the proposed deep residual network for frame rate up conversion. Section IV presents the proposed deep residual network with weighted convolutional motion compensation for frame rate up conversion. Section V provides experimental results. Finally, Section VI concludes the paper.

II. Related work

A. Motion-Compensated Frame Rate Up-Conversion

Many ME algorithms have been proposed to improve the accuracy of obtained motion vectors in FRUC. Hill et al. [25] proposed shape adaptive phase correlation to decrease the influence of background and predigest the identification of the dominant peak. Compared with the traditional full-search BMA, a novel adaptive fast BMA [26] was proposed to determine the motion vectors with a wide range of motion content. Kang et al. [16] proposed the dual motion estimation (Dual ME) to refine the motion vectors using the unidirectional and bidirectional matching ratios of blocks in the previous and following frames. The extended bilateral motion estimation (EBME) [13] employed expanded motion trajectory to calculate motion vectors of the block overlapping with each of two adjacent original blocks and then utilize motion vector refinement and smoothing in order to find the accurate motion vectors. With their direction-select motion estimation (DS-ME) [14], Yoo et al. performed motion estimation in both forward and backward directions and then selected the more reliable one from them. Li et al. [15] incorporated multi-resolution search into BME and constructed the wavelet pyramid to reduce computational cost and reserve the high-frequency details of interpolated frame. However, the above-mentioned algorithms may induce blocking artifacts due to incorrect motion vectors.

The existing motion compensated interpolation algorithm in FRUC mainly includes the overlapped block motion compensation (OBMC) [19], [20], [21], spatio-temporal auto-regressive (STAR) [27], [28], the weighted index-based bidirectional MCI (WMCI) [13], the direction-select ME based OBMC algorithm (DS-OBMC) [14], and the dual-weighted overlapped block motion compensation (DW-OBMC) [29]. OBMC and DS-OBMC use the mean value of forward and backward interpolated frame candidates as the interpolated frame. WMCI and DW-OBMC define the weighted coefficients of the previous and following interpolated frame candidates by calculating the sum of overlapped area absolute difference (SOAD). However, these MCI algorithms [13], [19], [14] still fail to calculate the optimal weighted coefficients.

B. Convolutional Neural Network

Recently, convolutional neural network (CNN) [30] has achieved great success in representation of image features [31] and image classification. Krizhevsky et al. [32] trained a large, deep convolutional neural network to classify massive images in the ImageNet and greatly reduce the error rate. Yim et al. [33] extracted features from multiple layers to achieve the improved performance. CNN is also applied in generating high-quality images. Dong et al. [34] proposed a three layer convolutional neural network to improve the performance of super-resolution. Kim et al. [35] verified that the deeper convolutional network is trained, the better performance is achieved. Zhang et al. [36] proposed a residual highway convolutional neural network to improve the quality of reconstructed frame in HEVC. In [37], an adaptive residual network was proposed for image restoration. Meanwhile, training deeper CNN widens the receptive field [35], which is applicable to image restoration problems. However, the deep network architectures are hard to train because of a mass of parameters and limited training dataset. To mitigate the problem of over-abundant parameters, Kim et al. [38] proposed deeply-recursive convolutional network (DRCN) for super-resolution. Deeply recursive residual network (DRRN) [39]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

was also proposed to improve the peformance of recovered image. These CNN based methods generate an image with high quality by reconstructing realistic texture detail.

III. Deep Residual Network for Frame Rate Up Conversion

A. Overall Architecture of the Network

Fig. 1 shows the overall architecture of the proposed deep residual network for frame rate up conversion (DRNFRUC). We firstly produce an initial interpolated frame defined as I_n^F by using an arbitrary existing FRUC algorithm. Then, I_n^F is fed into our deep residual network. Finally, the output of the network is defined as $\hat{I}_n$. DRNFRUC is formulated as follows:

$$\begin{aligned} I_n^F &= FRUC(I_{n-1}, I_{n+1}), \\ \hat{I}_n &= DRN(I_n^F), \end{aligned} \tag{1}$$

where function $FRUC$ denotes the process of FRUC using the previous frame I_{n-1} and the following frame I_{n+1} as input, function DRN denotes the deep residual network enhancing the interpolated frame I_n^F.

B. Deep Residual Network

Following Ledig et al. [40], our deep residual network shown in Fig. 2 consists of three parts. The first part, feature extraction, uses convolution filters of which the size is 3×3 to extract features of the image as feature maps. Then, we add batch normalization layer followed by ReLU, which acts as the activation function, in order to decrease training time.

The second part called feature recursive analysis widens the receptive field to analyze image feature extracted from larger image region with each recursion. For each recursion, we use a residual block [41], [42] illustrated in Fig. 3:

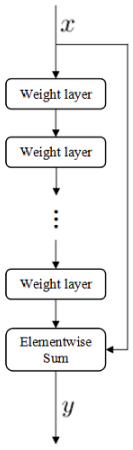


Fig. 3: Architecture of residual block

$$y = x + H(x), \tag{2}$$

where x and y denote the input and output of the residual block, respectively, H denotes the mapping of the residual block. We also use 3×3 convolution kernels for each filter. The third part is called image restoration which uses the output of the feature recursive analysis to obtain the interpolated frame.

TABLE I: Validation dataset

video	resolution	video	resolution
BlowingBubbles	416×240	NewspaperR0	1024×768
NightMove	704×576	Silent	352×288

This part only uses 1 filter with 3×3 convolution kernel because we merely concentrate on one luminance channel for each frame. Besides these operations, we add the skip connection between the input of deep residual network and the output of image restoration. We consider the three end-to-end parts as a mapping. Then, the input of deep residual network is directly transmitted to the output of the mapping, similar to a big residual block. The skip connection not only has fast convergence but helps our algorithm obtain better solution.

C. Analysis of Structure and Number of Residual Block

To select the optimal filter number, the structure of the residual blocks, and the number of the residual blocks, we choose DS-ME to conduct the simulation. Both the training and testing sequences are compressed by the reference software JM 18.6 at quantization parameter QP=32. 4 video sequences are chosen as the validation dataset presented in Table I to select optimal parameters of the proposed DRNFRUC.

1) Structure of Residual Block: We fist set filter number $n_1 = 64$ in the feature extraction part and number of residual blocks $c = 16$ in the feature recursive analysis part inspired by Kim [35] and Ledig [40] . For each residual network, we use convolution layers, batch normalization layers and ReLU. In general, the performance would improve if we design the reasonable structure of residual block. We conduct three experiments with different structures of residual block shown in Fig. 4: (a) each residual block contains one convolutional layer followed by one batch normalization layer defined as Cov+BN, (b) each residual block contains two convolutional layers, two batch normalization layers and one activation function defined as 2Cov+2BN+ReLU, (c) each residual block contains three convolutional layers, three batch normalization layers and two activation functions defined as 3Cov+3BN+2ReLU. The PSNR values over the test sequences observed at 3.5×10^5 backpropagations are shown in Table II. It is clear that superior performance could be achieved by using 2Cov+2BN+ReLU. The result shows that a uncomplicated residual network is preferred when it is uncertain that adding more weights and increasing the complexity of residual network would improve the performance.

2) Filters and Number of Residual Blocks in Deep Residual Network: In He et al.'s work [41] , the depth of network greatly influences the performance. In the pioneering work of Dong et al. [34] , adding more filters increases the performance moderately at the cost of running time. Following above-mentioned experiments, we adopt the 2Cov+2BN+ReLU as the basic structure of residual network. To validate the influence of changing the number of layers and filters, we choose $c = 8, 16, 20$ and $n_1 = 32, 64, 96$, i.e., we conduct a total of 9 experiments with different combinations in DRNFRUC. After 3.3×10^5 iterations, the average PSNR values over the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

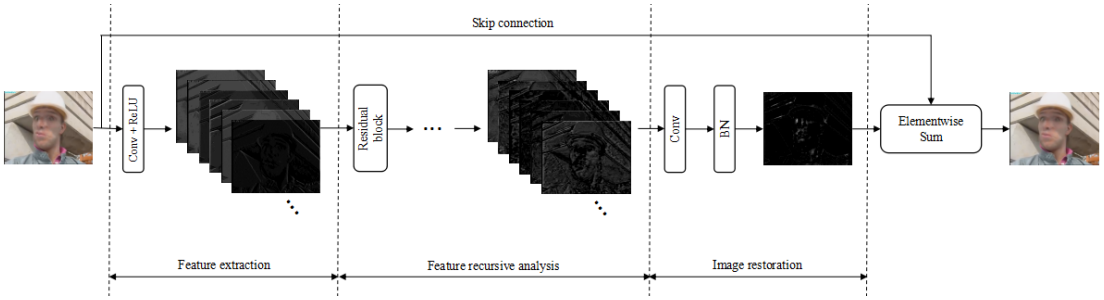


Fig. 2: Architecture of deep residual network

TABLE II: Average PSNRs on all the validation dataset with different structures of residual blcck at QP=32

residual block / video sequences	Cov +BN	2Cov +2BN+ReLU	3Cov +3BN+2ReLU
BlowingBubbles	29.92	30.04	30.03
NewspaperR0	36.75	36.91	36.93
NightMove	26.21	26.20	26.21
Silent	31.80	31.85	31.84
Average	**31.17**	**31.25**	**31.25**

TABLE III: Average PSNRs on all validation dataset of different filter number and number of residual blocks for deep residual network at QP=32

Number of residual blocks (c) / Filter number (n_1)	8	16	20
32	31.23	31.24	31.24
64	31.24	31.25	31.26
96	31.24	31.23	31.25

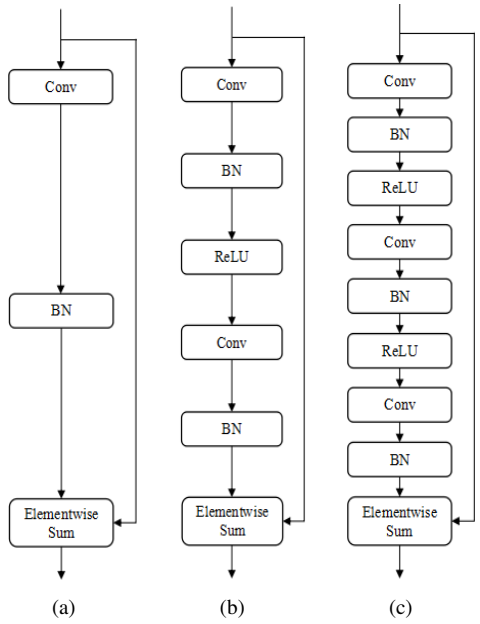


Fig. 4: Structure of resudual block: (a) Cov+BN (b) 2Cov+2BN+ReLU (c) 3Cov+3BN+2ReLU

test sequences are shown in Table III. The result shows that it is not necessary to employ more filters and more residual blocks to improve the performance. When we set $c = 16$ and $n_1 = 64$, the performance of the network is slightly higher than the others.

IV. Deep Residual Network with Weighted Convolutional Motion Compensation for Frame Rate Up Conversion

Since DRNFRUC only employs the interpolated frame generated by the existing FRUC algorithms, neglecting the influence of motion estimation and motion compensation interpolation in FRUC, we also propose a deep residual network with weighted convolutional motion compensation (DRNWCMC) in this Section to further improve the performance of FRUC.

A. Overall Scheme of DRNWCMC Algorithm

Our proposed algorithm shown in Fig 5. consists of three components. We firstly obtain forward motion compensated frame I_{vf} and backward motion compensated frame I_{vb} by using bilateral motion estimation. It should be noted that other methods can also be employed to estimate I_{vf} and I_{vb}. Then, we use weighted convolutional motion compensated interpolation (WCMCI) to generate an initial interpolated frame I_n^{WC}. Finally, a deep residual network is trained to reduce blocking artifacts and enhance the quality of interpolated frame. The whole process of DRNWCMC is formulated as:

$$\begin{aligned} I_n^{WC} &= WCMC(I_{vf}, I_{vb}), \\ \hat{I}_n &= DRN(I_n^{WC}), \end{aligned} \tag{3}$$

where function $WCMC$ denotes the process of WCMCI.

B. Forward and Backward Convolutional Neural Network

When the motion is violent, it is very difficult to obtain I_{vf} and I_{vb} with high accuracy, resulting in interpolated frame with low quality. To address this problem, we use forward and backward convolutional neural networks to improve the quality of I_{vf} and I_{vb}, respectively. Forward and backward convolutional neural networks have the same structure as shown in Fig. 6. Both of the two networks use d convolution+ReLU layers and a convolution layer, which extract features by sharing the same filter weights. The parameter setting will be further

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

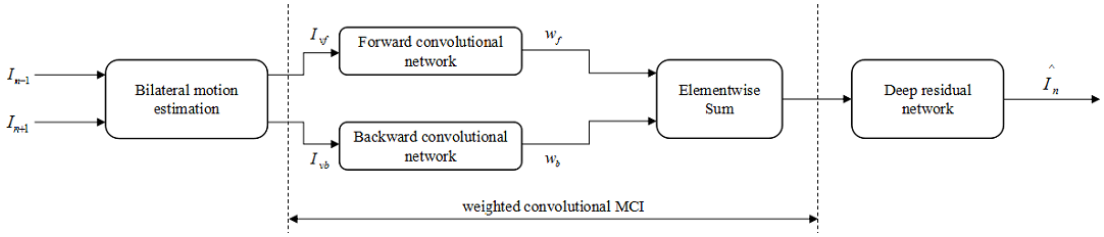


Fig. 5: DRNWCMC

discussed later in the following subsections. For each convolution+ReLU layer, we use n_2 filters with 3×3 convolution kernels. The last layer consists of a single filter with 3×3 convolutional kernels. We also add a skip connection between the input and output of both networks. The experiments show that the skip connection can reduce the convergence time and enhance the performance of the network.

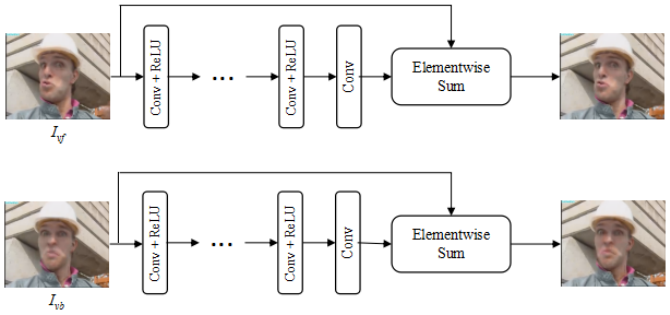


Fig. 6: Forward and backward convolutional neural network

C. Weighted Convolutional Motion Compensation Interpolation

Different from OBMC and DS-OBMC, we propose a novel weighted convolutional motion compensation interpolation (WCMCI) to improve the performance of motion compensation. The interpolation is defined as

$$I_n^{WC} = \frac{w_f \times D_f(I_{vf}) + w_b \times D_b(I_{vb})}{w_f + w_b}, \tag{4}$$

where functions D_f and D_b denote forward and backward CNNs, respectively, w_f and w_b are trainable variables. Our model consists of two stages: extended information filtering and bilateral weighted average reconstruction. The first stage employs D_f and D_b to enhance pixel information in I_{vf} and I_{vb} , respectively. Calculating the weighted average of the output of D_f and D_b, the second stage generates an interpolated frame I_n^{WC}. w_f and w_b, which are learned together with forward and backward convolutional neural networks, are viewed as a filter with 1×1 convolutional kernel, respectively.

D. Analysis of Filters and Number of layers in WCMCI

To obtain the appropriate number of filters and layers in WCMCI, we set $d = 8, 16, 20$ and $n_2 = 32, 64, 96$ during experiments. The result is shown in Table IV. We find that different number of filters and residual blocks in WCMCI do not have much impact on the objective evaluation.

TABLE IV: Average PSNRs on all validation dataset of different filter number and number of layers for DRNWCMC at QP=32

Filter number (n_2) \ Number of layers (d)	8	16	20
32	31.25	31.25	31.27
64	31.25	31.26	31.25
96	31.24	31.27	31.25

E. Loss Function

In order to obtain the optimal result, we minimize the mean squared error (MSE) between $\hat{I}_n$ and the ground truth I_n . Moreover, we add the quadratic sum of the weights of all the convolutional layers in the network to the loss function as the regularization term. We define the loss function as follows:

$$\begin{aligned} L_\theta &= \frac{1}{MNS} \sum_{n=1}^{S} \sum_{x=1}^{M} \sum_{y=1}^{N} (\hat{I}_n(\theta, x, y) - I_n(x, y))^2 \\ &\quad + \beta \sum \theta^2, \end{aligned} \tag{5}$$

where M and N denote the height and width of each frame, respectively. S denotes the number of frames during training, β denotes the coefficient of weight decay, and θ denotes the learnable weights of all the convolutional layers in the proposed network. If we define $D = \frac{1}{MNS} \sum_{n=1}^{S} \sum_{x=1}^{M} \sum_{y=1}^{N} (\hat{I}_n(\theta, x, y) - I_n(x, y))^2$, the gradient of the loss function in Eq. (5) can be expressed as

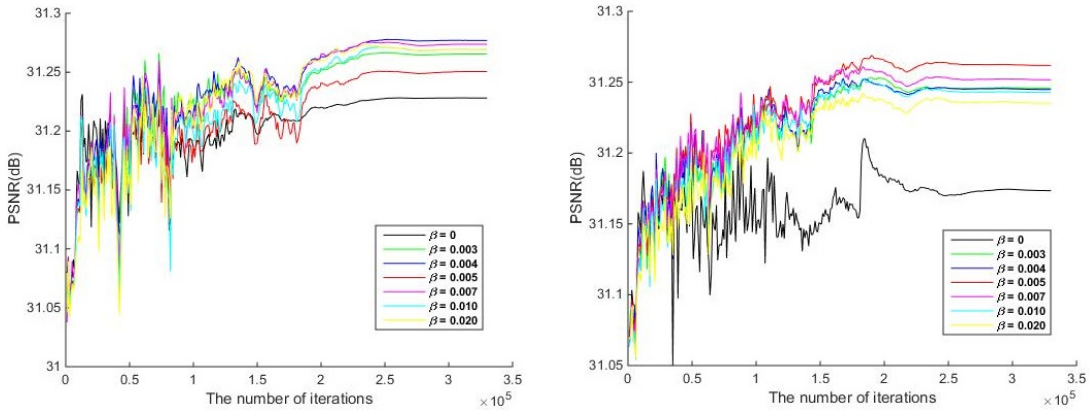
$$\frac{\partial L}{\partial \theta} = \frac{\partial D}{\partial \theta} + \beta\theta. \tag{6}$$

Therefore, the update process of the weights of the proposed network can be expressed as

$$\begin{aligned} \theta^{t+1} &= \theta^t - \eta \frac{\partial L}{\partial \theta} \\ &= \theta^t - \eta \left(\frac{\partial D}{\partial \theta} + \beta\theta^t \right) \\ &= (1 - \eta\beta)\theta^t - \eta \frac{\partial D}{\partial \theta}. \end{aligned} \tag{7}$$

We can see with the iteration going on, $(1-\eta\beta)\theta^t$ becomes closer to zero, which means multiplying the parameter θ by a factor smaller than one before the updating. It should be noted that the loss function in Eq. (5) is suitable to both of the two networks we proposed in this paper. And in DRNWCMC, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology



(a) (b)

Fig. 7: The PSNRs of validation dataset with different β during training: (a) DRNFRUC, (b) DRNWCMC

parameter θ includes the weights in the forward and backward convolutional neural network, the coefficients of w_f and w_b in Eq. (4), and the weights of the following deep residual network. In the experiment, we found that the performance of the network can be improved when giving an appropriate value of β. We conducted experiments to compare the results of the different β. Here, we set $c = 16$, $d = 16$, $n_1 = 64$ and $n_2 = 64$. The PSNRs of validation dataset with different β during training are shown in Fig. 7. We can see that DRNFRUC and DRNWCMC converge the fastest and gains the highest PSNR over the validation dataset when we set $\beta = 0.004$ and $\beta = 0.005$.

V. Experimental Results

In our experiments, all the sequences are compressed by the reference software JM18.6 of H.264 before performing FRUC. It should be noted that other coding standards can also be utilized. Since our method can be applied to all the FRUC algorithms, we only select DS-ME and EBME which use DS-OBMC and WMCI respectively for convenience. We use both the subjective and objective criterions to evaluate the quality of interpolated frames.

A. Training and testing Datasets

We choose 20 video sequences with different resolutions as the training dataset presented in Table V. These video sequences are compressed by JM18.6 under QP of 28, 32, 36, and 40. The validation dataset is shown in Table I, which is the same as the simulation in the Section III.C and Section IV.D. Besides, we choose 11 video sequences with different resolutions as the testing dataset presented in Table VI.

B. Training Parameters Setting

We use the aforementioned training dataset to train all the networks on a NVIDIA TITAN X. Each frame in the training dataset is cut into sub images. We randomly select 16 sub images as a mini-batch. Adam algorithm is used as optimization method to update the weights of the network. The initial learning rate is set to 0.0001 and decreases gradually during the training process. After 330 thousand update iterations, the results of validation set tend to converge.

TABLE V: Training dataset

video	resolution	video	resolution
BasketballDrill	832 × 480	BasketballDrillText	832 × 480
BasketballDrive	1920 × 1080	BQMall	832 × 480
BQTerrace	1920 × 1080	Cactus	1920 × 1080
ChinaSpeed	1024 × 768	ElephantsDream	352 × 288
FourPeople	1280 × 720	Johnny	1280 × 720
Kimono1	1920 × 1080	KristenAndSara	1280 × 720
ParkScene	1920 × 1080	PartyScene	832 × 480
PeopleOnStreet	2560 × 1600	RaceHorses	832 × 480
Traffic	2560 × 1600	Vidyo1	1280 × 720
Vidyo3	1280 × 720	Vidyo4	1280 × 720

TABLE VI: Testing dataset

video	resolution	video	resolution
Akiyo	352 × 288	BigShips	1280 × 720
Container	352 × 288	Crew	1280 × 720
Foreman	352 × 288	Hall	352 × 288
Harbour	1280 × 720	Mobile	352 × 288
Mobisode2	416 × 240	Mother-daughter	352 × 288
News	352 × 288		

C. Subjective and Objective Evaluation

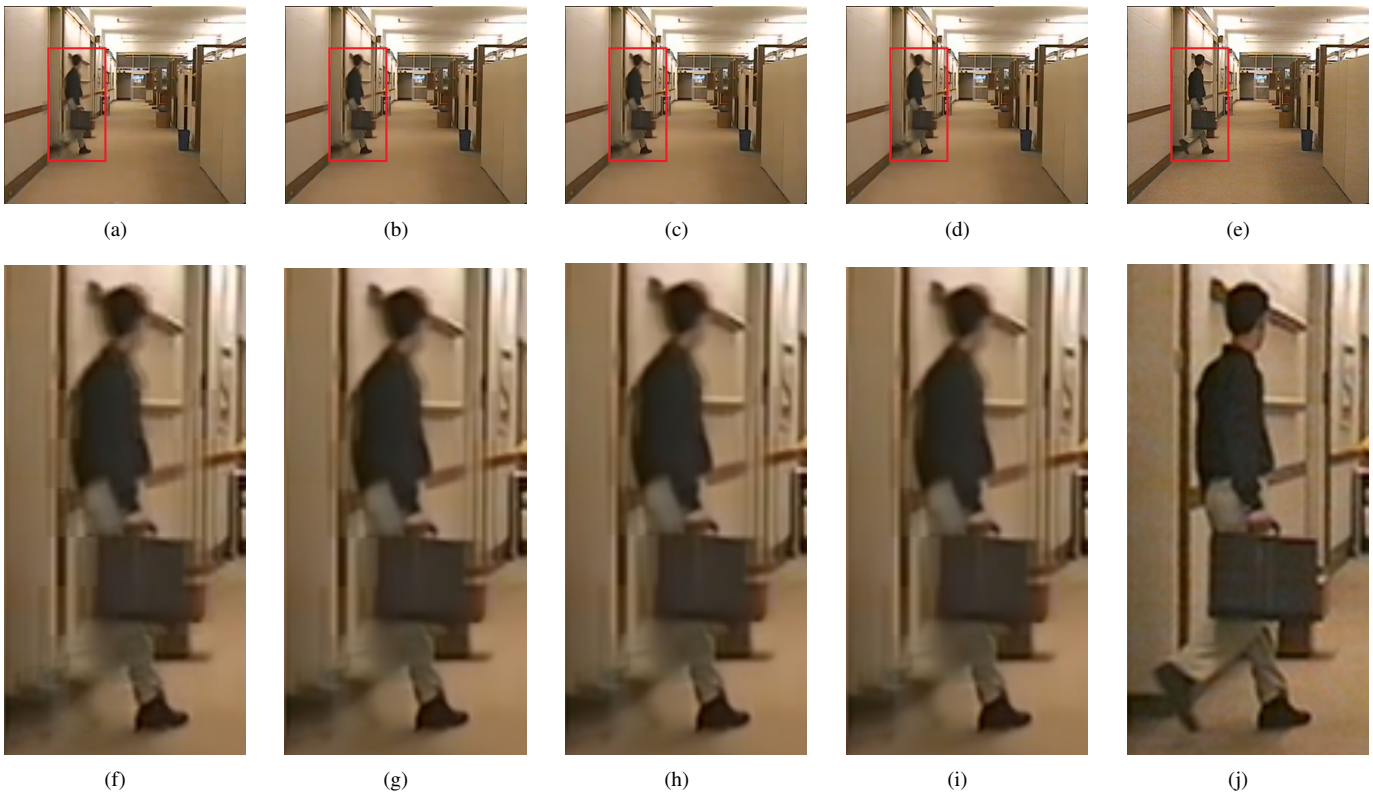
Here, we select $c = 16$, $d = 16$, $n_1 = 64$, $n_2 = 64$ and 2Cov+2BN+ReLU as the parameters of the network in order to not only keep superior performance but also generate a high quality interpolated frame at low computation cost. Firstly, we compare the performance of selected FRUC algorithm with corresponding DRNFRUC ($\beta = 0$), DRNFRUC ($\beta = 0.005$) and DRNWCMC ($\beta = 0.005$) from subjective criterion view. In Fig. 8 and Fig. 9, we visually assess some interpolated frames generated by DS-ME and our proposed algorithms over Foreman and Hall at QP=32. We can find that DS-ME produces different levels of blurring and blocking artifacts when the motion of the block is violent. The deformation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 8: The 79 th interpolated frames of Foreman reconstructed by different FRUC algorithms: (a) and (f) interpolated by DS-ME, (b) and (g) interpolated by DRNFRUC ($\beta = 0$), (c) and (h) interpolated by DRNFRUC ($\beta = 0.005$), (d) and (i) interpolated by DRNWCMC, (e) and (j) original image



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 9: The 32 th interpolated frames of Hall reconstructed by different FRUC algorithms: (a) and (f) interpolated by DS-ME, (b) and (g) interpolated by DRNFRUC ($\beta = 0$), (c) and (h) interpolated by DRNFRUC ($\beta = 0.005$), (d) and (i) interpolated by DRNWCMC, (e) and (j) original image

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

TABLE VIII: PSNRs and SSIMs of the interpolated frames reconstructed by the proposed and DS-ME algorithms for testing dataset at QP=24, 28, 32

Sequence	DS-ME			DRNFRUC ($\beta = 0$)			DRNFRUC ($\beta = 0.005$)			DRNWCMC ($\beta = 0.005$)		
	QP=24	QP=28	QP=32	QP=24	QP=28	QP=32	QP=24	QP=28	QP=32	QP=24	QP=28	QP=32
Akiyo	41.78 0.9792	39.87 0.9689	37.47 0.9559	41.47 0.9585	40.00 0.9700	37.90 0.9593	41.57 0.9791	40.04 0.9703	37.97 0.9598	41.77 0.9797	40.17 0.9709	38.09 0.9602
BigShips	36.66 0.9532	35.47 0.9843	33.66 0.9683	36.83 0.9546	35.72 0.9848	34.00 0.9699	36.82 0.9545	35.71 0.9848	34.02 0.9700	36.85 0.9547	35.70 0.9849	34.04 0.9702
Container	38.46 0.9596	35.96 0.9208	33.53 0.8871	38.25 0.9594	35.92 0.9214	33.57 0.8884	38.31 0.9595	35.96 0.9217	33.61 0.8887	38.40 0.9601	36.01 0.9219	33.66 0.8889
Crew	33.75 0.9236	33.47 0.9538	32.86 0.9445	33.87 0.9256	33.58 0.9549	33.03 0.9464	33.89 0.9264	33.63 0.9555	33.06 0.9468	33.91 0.9265	33.63 0.9555	33.09 0.9475
foreman	32.63 0.9279	32.03 0.9025	31.00 0.8756	32.61 0.9278	32.09 0.9032	31.14 0.8787	32.66 0.9281	32.13 0.9041	31.17 0.8793	32.65 0.9281	32.17 0.9045	31.20 0.8798
Hall	36.64 0.9628	36.03 0.9489	34.71 0.9385	36.60 0.9634	36.08 0.9508	34.85 0.9417	36.62 0.9635	36.07 0.9507	34.91 0.9419	36.69 0.9637	36.12 0.9508	34.92 0.9418
Harbour	34.11 0.9557	33.73 0.9871	32.61 0.9817	34.23 0.9578	33.89 0.9874	32.82 0.9825	34.24 0.9579	33.90 0.9875	32.85 0.9826	34.22 0.9577	33.89 0.9874	32.85 0.9826
Mobile	28.82 0.9522	28.44 0.9415	27.64 0.9218	29.04 0.9548	28.69 0.9451	27.92 0.9269	29.04 0.9546	28.68 0.9449	27.92 0.9268	29.03 0.9546	28.68 0.9447	27.90 0.9262
Mobisode2	41.34 0.9711	40.10 0.9633	38.46 0.9531	40.90 0.9688	39.96 0.9621	38.44 0.9530	40.98 0.9702	40.05 0.9631	38.52 0.9544	41.24 0.9712	40.12 0.9641	38.65 0.9551
Mother-daughter	40.32 0.9711	38.49 0.9510	36.46 0.9258	40.22 0.9698	38.65 0.9527	36.73 0.9302	40.33 0.9703	38.72 0.9533	36.75 0.9308	40.43 0.9708	38.77 0.9537	36.81 0.9309
News	34.57 0.9623	33.89 0.9505	32.76 0.9359	34.61 0.9628	34.02 0.9522	33.04 0.9395	34.64 0.9629	34.05 0.9526	33.06 0.9401	34.67 0.9635	34.07 0.9529	33.07 0.9401
Average	**36.28** **0.9561**	**35.23** **0.9521**	**33.74** **0.9353**	**36.24** **0.9548**	**35.33** **0.9531**	**33.95** **0.9379**	**36.28** **0.9570**	**35.36** **0.9535**	**33.99** **0.9383**	**36.35** **0.9573**	**35.39** **0.9538**	**34.03** **0.9385**

TABLE VII: PSNRs of the interpolated frames reconstructed by DS-ME, WCMC-FRUC and DRNWCMC for testing dataset at QP=28, 32

Sequence	DS-ME		WCMC-FRUC ($\beta = 0.005$)		DRNWCMC ($\beta = 0.005$)	
	QP=28	QP=32	QP=28	QP=32	QP=28	QP=32
Akiyo	39.87	37.47	39.91	37.51	40.17	38.09
BigShips	35.47	33.66	35.51	33.71	35.70	34.04
Container	35.96	33.53	36.00	33.56	36.01	33.66
Crew	33.47	32.86	33.50	32.91	33.63	33.09
foreman	32.03	31.00	32.04	31.01	32.17	31.20
Hall	36.03	34.71	36.03	34.71	36.12	34.92
Harbour	33.73	32.61	33.75	32.65	33.89	32.85
Mobile	28.44	27.64	28.45	27.66	28.68	27.90
Mobisode2	40.10	38.46	40.12	38.48	40.12	38.65
Mother-daughter	38.49	36.46	38.54	36.51	38.77	36.81
News	33.89	32.76	33.90	32.77	34.07	33.07
Average	**35.23**	**33.74**	**35.25**	**33.77**	**35.39**	**34.03**

and blocking effects appear in the red rectangle of the interpolated frame. However, DRNFRUC effectively eases these problems and reconstructs a high quality interpolated frame. From subjective perspective view, the interpolate frames by DRNWCMC are more distinct than those of DRNFRUC.

Besides, we conduct experiments to evaluate the result of WCMC-FRUC. The PSNRs of DS-ME and our proposed algorithms on each testing sequence are presented in Table VII. It can be observed WCMC alone outperforms DS-ME by up to 0.02dB and 0.03dB at QP=28 and 32, respectively. Obviously, the gain is not very significant by WCMC alone, which proves the necessity of the convolutional neural network following WCMCI in the proposed DRNWCMC.

Then, we evaluate the objective quality of our proposed algorithms and DS-ME in terms of PSNR and structural similarity index metric (SSIM) [43] which are popular evaluation criterions of images and videos. The PSNRs and SSIMs of our proposed algorithms and DS-ME on each testing sequence at different QPs are presented in Table VIII and Table IX. It can be observed that when compared with DS-ME, the proposed algorithms improve the PSNRs by up to 0.22dB, 0.30dB, 0.62dB, 0.74dB and 0.64dB, at QP=24, 28, 32, 36, 40, respectively. The average PSNRs of different algorithms on all the testing sequences are presented in the last row of each Table, and the proposed algorithms outperform DS-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

TABLE IX: PSNRs and SSIMs of the interpolated frames reconstructed by the proposed and DS-ME algorithms for testing dataset at QP=36, 40

Sequence	DS-ME		DRNFRUC ($\beta = 0$)		DRNFRUC ($\beta = 0.005$)		DRNWCMC ($\beta = 0.005$)	
	QP=36	QP=40	QP=36	QP=40	QP=36	QP=40	QP=36	QP=40
Akiyo	35.01 0.9375	32.42 0.9099	35.74 0.9444	33.02 0.9175	35.73 0.9446	33.06 0.9179	35.75 0.9444	33.06 0.9185
BigShips	31.69 0.9383	29.53 0.8842	32.10 0.9418	29.92 0.8902	32.09 0.9416	29.92 0.8903	32.11 0.9418	29.93 0.8904
Container	31.19 0.8560	28.87 0.8249	31.40 0.8591	29.18 0.8314	31.40 0.8591	29.22 0.8320	31.45 0.8591	29.23 0.8312
Crew	32.01 0.9284	30.85 0.8998	32.19 0.9314	31.01 0.9048	32.22 0.9320	31.04 0.9055	32.26 0.9329	31.05 0.9058
Foreman	29.84 0.8405	28.37 0.7939	30.08 0.8466	28.68 0.8026	30.08 0.8468	28.68 0.8028	30.10 0.8468	28.69 0.8026
Hall	32.77 0.9216	30.37 0.8946	33.07 0.9281	30.85 0.9056	33.10 0.9278	30.87 0.9061	33.12 0.9281	30.85 0.9054
Harbour	30.85 0.9680	28.60 0.9368	31.12 0.9698	28.91 0.9408	31.14 0.9698	28.91 0.9407	31.17 0.9701	28.93 0.9409
Mobile	26.44 0.8866	24.58 0.8150	26.78 0.8948	24.92 0.8267	26.76 0.8941	24.93 0.8267	26.75 0.8939	24.91 0.8259
Mobisode2	36.70 0.9392	34.68 0.9211	36.82 0.9407	34.91 0.9245	36.84 0.9411	34.96 0.9255	36.98 0.9424	35.07 0.9267
Mother-daughter	34.28 0.8922	31.95 0.8482	34.65 0.8985	32.17 0.8566	34.66 0.8990	32.17 0.8568	34.71 0.8997	32.26 0.8577
News	31.23 0.9134	29.29 0.8789	31.64 0.9200	29.79 0.8897	31.69 0.9205	29.81 0.8897	31.69 0.9207	29.80 0.8893
Average	**32.00** **0.9110**	**29.96** **0.8734**	**32.33** **0.9159**	**30.31** **0.8809**	**32.34** **0.9160**	**30.32** **0.8813**	**32.37** **0.9164**	**30.34** **0.8813**

ME by up to 0.07dB, 0.16dB, 0.29dB, 0.37dB and 0.38dB, at QP=24, 28, 32, 36, 40, respectively. The average SSIMs of different algorithms on all the testing sequences are also presented in the last row of each Table, and it shows that the proposed algorithms outperform DS-ME by up to 0.0029, 0.0017, 0.0032, 0.0054 and 0.0079, at QP=24, 28, 32, 36, 40, respectively. Fig. 10 shows the PSNRs of individual interpolated frames on the Akiyo, BigShips, Container, and Hall sequences at QP=32. These PSNR curves prove that the proposed algorithms outperform DS-ME in most cases.

To further verify the applicability of the proposed algorithms to all the FRUC algorithms, we also conduct the experiments over EBME. The average PSNRs of our proposed algorithms and EBME on all the testing sequences at different QPs are presented in Table X, and it can be seen that the proposed algorithms also gain up to 0.05dB, 0.24dB, 0.28dB, 0.36dB, 0.40dB, at QP=24, 28, 32, 36, 40, respectively. In addition, Table VIII summarizes the average SSIMs on all the testing sequences and shows that the average SSIM of our proposed algorithms is up to 0.0082 higher than EBME at QP=40.

From these Tables, we can find that the average PSNRs on all the testing sequences of DRNWCMC ($\beta = 0.005$) are up to 0.08dB and 0.04dB higher than those of DRNFRUC with $\beta = 0$ and those of DRNFRUC with $\beta = 0.005$, respectively, when DS-ME is selected as the basic FRUC algorithm. Meanwhile when EBME is used to process the compressed testing dataset, DRNWCMC ($\beta = 0.005$) improves the average PSNRs by up to 0.09dB and 0.10dB, respectively, compared to DRNFRUC with $\beta = 0$ and DRNFRUC with $\beta = 0.005$. The experimental results verify the superiority of WCMCI in both objective and subjective criterions compared with OBMC, DS-OBMC and WMCI.

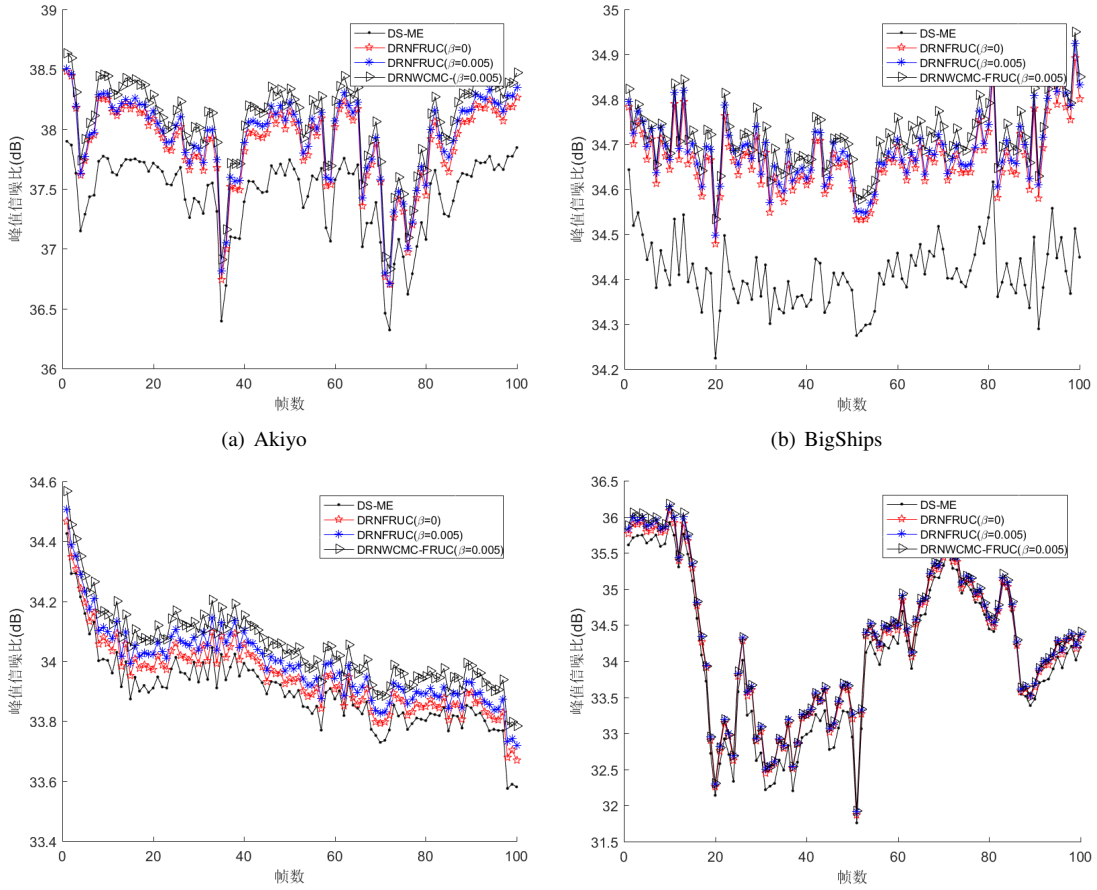
D. Performance on Expanding Training Dataset

We further investigate the performance of our proposed algorithms when we select more training sequences. Table XI shows 26 extra video sequences as added training sequences. We only evaluate the average PSNRs and SSIMs of our proposed algorithms on all the testing sequences at QP=32 shown in Table XII. The average PSNRs and SSIMs of our proposed algorithms using the expanded (46) training sequences are 0.12dB, 0.08dB, and 0.08dB higher than those of corresponding algorithms by using only 20 training sequences. Thus, expanding training dataset dramatically improves the performance of our proposed algorithms.

E. Computational Complexity Analysis

The average running times of traditional FRUC algorithms, DRNFRUC and DRNWCMC over a typical computer (Intel Core i7-5930K CPU 3.50 GHz, 16 GB memory and NVIDIA TITAN X) are shown in Table XIII. Obviously, the running

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology



(a) Akiyo (b) BigShips

(c) Container (d) Hall

Fig. 10: PSNRs of individual interpolated frames reconstructed by the proposed and conventional algorithms for Akiyo, BigShips, Container, and Hall sequences at QP=32

TABLE X: Average PSNRs and SSIMs on all the testing dataset reconstructed by the proposed and EBME algorithms at different QP

QP	EBME	DRNFRUC ($\beta = 0$)	DRNFRUC ($\beta = 0.005$)	DRNWCMC ($\beta = 0.005$)
24	36.21 / 0.9544	36.08 / 0.9542	36.14 / **0.9565**	**36.26** / 0.9550
28	35.26 / 0.9441	35.32 / 0.9454	35.38 / 0.9457	**35.50** / **0.9456**
32	33.74 / 0.9257	33.92 / 0.9285	33.95 / 0.9289	**34.02** / **0.9291**
36	32.00 / 0.8997	32.27 / 0.9043	32.30 / 0.9050	**32.36** / **0.9051**
40	29.95 / 0.8610	30.29 / **0.8692**	30.28 / 0.8688	**30.35** / 0.8690

times of our proposed algorithms are longer than that of traditional FRUC algorithms. Besides, we apply the GPU parallel computing to speed up the process of generating the interpolated frame. The running times of our proposed algorithms reduce to a tolerable limit with GPU parallel computing, which contributes to practical usage of our proposed algorithms.

VI. Conclusions

In this paper, we propose two novel FRUC algorithms to improve the performance of interpolated frames. A deep residual neural network for frame rate up conversion (FRUC) is first devised to directly take the output of any exiting FRUC methods as the input and enhance the quality of the interpolated frames. To further improve the performance, we also proposed a deep residual network with weighted convolutional motion compensation (DRNWCMC) for FRUC. In DRNWCMC, the convolution operation can be embedded into the motion compensation interpolation process of traditional FRUC method, and consequently achieves superior performance. In addition, to improve the robustness of the proposed method, we also added the summation of the parameters of the network as the regularization term in the loss function. Experimental results show that the proposed algorithms have achieved superior performance than the traditional FRUC algorithms.

References

[1] Y. Guo, L. Chen, Z. Gao, and X. Zhang, "Frame Rate Up-Conversion Method for Video Processing Applications," *IEEE Transactions on Broadcasting*, vol. 60, no. 4, pp. 659–669, 2014.

[2] K. Hilman, H. W. Park, and Y. Kim, "Using motion-compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 10, no. 6, pp. 869–877, 2016.

[3] J. Hwang, Y. Choi, and Y. Choe, "Frame Rate Up-Conversion Technique Using Hardware-Efficient Motion Estimator Architecture for Motion Blur Reduction of TFT-LCD," *IEICE Transactions on Electronics*, vol. 94, no. 5, pp. 896–904, 2011.

[4] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2885564, IEEE Transactions on Circuits and Systems for Video Technology

TABLE XI: Expanded training dataset

video	resolution	video	resolution
Balloons1	1024×768	Cafec1	1920×1072
CrossStreet	704×576	Cyclists	1280×720
Dancerc1	1920×1088	GTFlyc1	1920×1088
Ice	704×576	Interview	704×576
Kendo1	1024×768	Lovebird1cam01	1024×768
Night	1280×720	NightStatic	704×576
Optis	1280×720	ParkScene	1920×1072
PoznanCarPark00	1920×1088	PoznanHall200	1920×1088
PoznanStreet00	1920×1088	Raven	1280×720
Sailormen	1280×720	Sheriff	1280×720
Soccer	704×576	Spincalendar	1280×720
SubStation	704×576	Subway	704×576
Tabletennis	704×576	Tennis	1920×1072

TABLE XII: The PSNRs of the interpolated frames reconstructed by the proposed algorithms for testing datasetat on expanding training dataset

video sequences	DRNFRUC ($\beta = 0$)	DRNFRUC ($\beta = 0.005$)	DRNWCMC ($\beta = 0.005$)
Akiyo	38.09	38.09	38.14
BigShips	34.18	34.14	34.19
Container	33.75	33.75	33.72
Crew	33.11	33.12	33.14
Foreman	31.23	31.23	31.27
Hall	35.06	35.04	35.09
Harbour	32.92	32.97	32.99
Mobile	27.96	27.98	27.97
Mobisode2	38.50	38.54	38.66
Mother-daughter	36.83	36.80	36.90
News	33.13	33.15	33.16
Average	**34.07**	**34.07**	**34.11**

TABLE XIII: Average running times of different algorithms per frame with respect to various resolutions

Algorithm \ Resolution	416×240	352×288	1280×720
DS-ME	0.35	0.35	3.49
DRNFRUC (CPU)	3.87	3.91	34.61
DRNWCMC (CPU)	6.22	6.47	56.81
DRNFRUC (GPU)	0.42	0.42	4.13
DRNWCMC (GPU)	0.5	0.46	4.47

[5] C. Cafforio, F. Rocca, and S. Tubaro, "Motion compensated image interpolation," *IEEE Transactions on Communications*, vol. 38, no. 2, pp. 215–222, 1990.

[6] S. J. Kang, D. G. Yoo, S. K. Lee, and Y. Kim, "Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1830–1838, 2008.

[7] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-Compensated Frame Rate Up-ConversionPart II: New Algorithms for Frame Interpolation," *IEEE Transactions on Broadcasting*, vol. 56, no. 2, pp. 142–149, 2010.

[8] B. W. Jeon, G. I. Lee, S. H. Lee, and R. H. Park, "Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 499–508, 2003.

[9] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Frame rate up-conversion using perspective transform," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 975–982, 2006.

[10] Y. Zhang, L. Xu, X. Ji, and Q. Dai, "A polynomial approximation motion estimation model for motion-compensated frame interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 8, pp. 1421–1432, 2016.

[11] B. D. Choi, J. W. Han, C. S. Kim *et al.*, "Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 17, no. 4, pp. 407–416, 2007.

[12] Y. Zhang, D. Zhao, S. Ma, R. Wang, and W. Gao, "A motion-aligned auto-regressive model for frame rate up conversion," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1248–1258, 2010.

[13] S.-J. Kang, K.-R. Cho, and Y. H. Kim, "Motion compensated frame rate up-conversion using extended bilateral motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, 2007.

[14] D. G. Yoo, S. J. Kang, and Y. H. Kim, "Direction-Select Motion Estimation for Motion-CompensatedFrame Rate Up-Conversion," *Journal of Display Technology*, vol. 9, no. 10, pp. 840–850, 2013.

[15] R. Li, H. Liu, J. Chen, and Z. Gan, "Wavelet Pyramid Based Multi-Resolution Bilateral Motion Estimation for Frame Rate Up-Conversion," *IEICE Transactions on Information and Systems*, vol. 99, no. 1, pp. 208–218, 2016.

[16] S. J. Kang, S. Yoo, and Y. H. Kim, "Dual Motion Estimation for Frame Rate Up-Conversion," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 20, no. 12, pp. 1909–1914, 2011.

[17] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4613–4626, 2013.

[18] X. Zhang, R. Xiong, W. Lin, S. Ma, J. Liu, and W. Gao, "Video compression artifact reduction via spatio-temporal multi-hypothesis prediction," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 6048–6061, 2015.

[19] T. Ha, S. Lee, and J. Kim, "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 752–759, 2004.

[20] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach." *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 3, no. 5, p. 693, 1994.

[21] B. T. Choi, S. H. Lee, and S. J. Ko, "New frame rate up-conversion using bi-directional motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 603–609, 2002.

[22] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for hevc," in *Data Compression Conference (DCC), 2017*. IEEE, 2017, pp. 410–419.

[23] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," in *Visual Communications and Image Processing (VCIP), 2017 IEEE*. IEEE, 2017, pp. 1–4.

[24] X. Song, J. Yao, L. Zhou, L. Wang, X. Wu, D. Xie, S. Pu *et al.*, "A practical convolutional neural network as loop filter for intra frame," *arXiv preprint arXiv:1805.06121*, 2018.

[25] L. Hill and T. Vlachos, "Motion measurement using shape adaptive phase correlation," *Electronics Letters*, vol. 37, no. 25, pp. 1512–1513, 2001.

[26] S. Y. Huang, C. Y. Cho, and J. S. Wang, "Adaptive fast block-matching algorithm by switching search patterns for sequences with wide-range motion content," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 15, no. 11, pp. 1373–1384, 2005.

[27] Y. Zhang, D. Zhao, X. Ji, R. Wang, and X. Chen, "A spatio-temporal autoregressive frame rate up conversion scheme," in *Proceedings of the IEEE Conference on Image Processing*, 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 Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 9, pp. 1289–1301, 2009.

[29] R. Li, Z. Gan, Z. Cui, G. Tang, and X. Zhu, "Multi-Channel Mixed-Pattern Based Frame Rate Up-Conversion Using Spatio-Temporal Motion Vector Refinement and Dual-Weighted Overlapped Block Motion Compensation," *Journal of Display Technology*, vol. 10, no. 12, pp. 1010–1023, 2014.

[30] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of

intelligent vehicles," *IEEE transactions on intelligent transportation systems*, vol. 19, no. 1, pp. 284–295, 2018.
[31] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang, and Q. Dai, "Effective uyghur language text detection in complex background images for traffic prompt identification," *IEEE transactions on intelligent transportation systems*, vol. 19, no. 1, pp. 220–229, 2018.
[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the Acm*, vol. 60, no. 2, p. 2012, 2012.
[33] J. Yim, J. Ju, H. Jung, and J. Kim, "Image classification using convolutional neural networks with multi-stage feature," in *Robot Intelligence Technology and Applications 3*. Springer, 2015, pp. 587–594.
[34] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199.
[35] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.
[36] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, "Residual Highway Convolutional Neural Networks for in-loop Filtering in HEVC," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3827–3841, 2018.
[37] Y. Zhang, L. Sun, C. Yan, X. Ji, and Q. Dai, "Adaptive Residual Networks for High-Quality Image Restoration," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3150–3163, 2018.
[38] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.
[39] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 4, 2017.
[40] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2016.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[42] K. He, X. Zhang, S. Ren *et al.*, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.
[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Yongbing Zhang received the B.A. degree in English and the M.S. and Ph.D degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively. He joined Graduate School at Shenzhen, Tsinghua University, Shenzhen, China in 2010, where he is currently an associate professor. He was the receipt of the Best Student Paper Award at IEEE International Conference on Visual Communication and Image Processing in 2015 and the Best Paper Award in Pacific Rim Conference on Multimedia 2018. His current research interests include signal processing, image/video coding, and machine learning.

Lixin Chen received the B.E. degree in electric engineering and automation from Hangzhou Dianzi University, China, in 2016. He is currently working toward the M.E. degree in Department of Automation, Hangzhou Dianzi University, China. His research interests include deep learning, video processing and video coding.

Chenggang Yan received the B.S. degree in Computer Science from Shandong University in 2008 and Ph.D. degree in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences in 2013. Now he is the Director of Intelligent Information Processing Lab in Hanzhou Dianzi Univeristy. Before that, he was an assistant research fellow in Tsinghua University. His research interests include intelligent information processing, machine learning, image processing, computational biology and computational photography.

He has authored or co-authored over 50 refereed journal and conference papers. As a co-author, he got the Best Paper Awards in Pacific Rim Conference on Multimedia 2018, International Conference on Internet Multimedia Computing and Service 2018, International Conference on Game Theory for Networks 2014, and SPIE/COS Photonics Asia Conference 2014, the Best Paper Candidate in International Conference on Multimedia and Expo 2011.

Peiwu Qin received the B.Sc. degree in biotechnology from Northeastern Agricultural University, Harbin, China, in 2002, the M.Sc. degree in developmental biology from Chinese Academy of Sciences, Beijing, China, in 2005, the M.Sc. degree in physical chemistry from Northeastern University, Boston, USA, in 2008, the M.Sc. degree in statistics from University of Missouri, Columbia, Missouri, USA, in 2013, and the Ph.D. degree in biochemistry from University of Missouri, Columbia, Missouri, USA, in 2013. He has been a Post-Doctoral Fellow with department of physics, University of California, Berkeley from 08/2013 to 08/2018. He is an assistant professor at Tsinghua Berkeley Shenzhen Institute, Shenzhen, China. His research interests include optical imaging and signal processing.

Xiangyang Ji received the B.S. degree in materials science and the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He joined Tsinghua University, Beijing, China, in 2008, where he is currently a Professor in the Department of Automation, School of Information Science and Technology.

His current research interests include signal processing, image/video processing, image/video compression and communication, 3-D representation, and reconstruction.

Qionghai Dai received the M.S. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively. He is currently a professor in the Department of Automation and is the Director of the Broadband Networks and Digital Media Laboratory at Tsinghua University, Beijing. He has authored or co-authored over 200 conference and journal papers and two books.

His research interests include computational photography and microscopy, computer vison and graphics, intelligent signal processing. He is associate Editor of JVCI, IEEE TNNLS and IEEE TIP.