Hindawi
Security and Communication Networks
Volume 2023, Article ID 9863209, 1 page
https://doi.org/10.1155/2023/9863209

Retraction

Retracted: Machine-Type Video Communication Using Pretrained Network for Internet of Things

Security and Communication Networks

Received 26 December 2023; Accepted 26 December 2023; Published 29 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] R. Li, P. Hao, F. Sun, Y. Li, and L. You, "Machine-Type Video Communication Using Pretrained Network for Internet of Things," *Security and Communication Networks*, vol. 2021, Article ID 6184797, 10 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6184797, 10 pages
https://doi.org/10.1155/2021/6184797

Research Article

Machine-Type Video Communication Using Pretrained Network for Internet of Things

Ran Li,[1,2] **Peinan Hao**,[1] **Fengyuan Sun**,[2] **Yanling Li**,[1] **and Lei You**[1]

[1]*School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China*
[2]*Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Ran Li; liran@xynu.edu.cn

Received 16 October 2021; Revised 15 November 2021; Accepted 23 November 2021; Published 6 December 2021

Academic Editor: Jian Su

With the increasing demand for internet of things (IoT) applications, machine-type video communications have become an indispensable means of communication. It is changing the way we live and work. In machine-type video communications, the quality and delay of the video transmission should be guaranteed to satisfy the requirements of communication devices at the condition of limited resources. It is necessary to reduce the burden of transmitting video by losing frames at the video sender and then to increase the frame rate of transmitting video at the receiver. In this paper, based on the pretrained network, we proposed a frame rate up-conversion (FRUC) algorithm to guarantee low-latency video transmitting in machine-type video communications. At the IoT node, by periodically discarding the video frames, the video sequences are significantly compressed. At the IoT cloud, a pretrained network is used to extract the feature layers of the transmitted video frames, which is fused into the bidirectional matching to produce the motion vectors (MVs) of the losing frames, and according to the output MVs, the motion-compensated interpolation is implemented to recover the original frame rate of the video sequence. Experimental results show that the proposed FRUC algorithm effectively improve both objective and subjective qualities of the transmitted video sequences.

1. Introduction

With the rapid development of the internet of things (IoT), more and more machines and autonomous devices are interconnected to produce various communication devices, such as smartphones, tablets, and set-top boxes. In the communication device, many visual sensors or cameras are used to capture the large-scale video data, and the video data are gathered in the cloud by a wireless network [1]. At the IoT nodes, due to the limited capacities of the storage and processing, it is difficult to provide the high-quality recovered video in real time [2], so it is necessary to reduce the frame rate of video at the IoT nodes to restrict the transmission rate. However, the video quality will be degraded seriously. To overcome this defect, some existing works tried to enhance the video quality by increasing the frame rate at the IoT cloud [3–5]. Therefore, it is challenging in a communication device to convert low-frame-rate video to high-frame-rate one. For example, to ensure the smooth running of the videoconferencing, it is a common method to reduce the frame rate of video at the nodes and increase the frame rate at the cloud.

Frame rate up-conversion (FRUC) refers to a technique that increases the frame rate of the transmitted video by exploiting the temporospatial correlations of adjacent frames [6]. It can improve the visual quality of the transmitted video, so some real-time applications use it to prevent the degradation of quality. Recently, FRUC has become a basic step to increase the frame rate of video in many IoT applications [7–9]. Therefore, many works have been proposed to develop effective FRUC algorithms [10–12].

FRUC is divided into two types including the motion-compensated FRUC (MC-FRUC) and non-MC-FRUC [13]. Non-MC-FRUC interpolates the absent frames by copying the previous frame or averaging the previous frame and the following frame, and it is suitable for low-speed videos. Non-MC-FRUC cannot generate satisfactory interpolated results due to

neglect of objective motions. MC-FRUC [14–16] exploits motion trajectories between adjacent frames to improve the interpolation quality, so it is commonly used to up-convert the video sequences with complex motions. MC-FRUC consists of motion estimation (ME) and motion-compensated interpolation (MCI). ME is used to calculate motion vectors (MVs) of interpolated frames, and MCI is used to interpolate the absent frames according to MVs output by ME [17]. The interpolation quality of MC-FRUC heavily depends on the ME accuracy, so the existing works focus on how to improve the implementation of ME. The block matching algorithm (BMA) is widely applied to ME due to its intuitive architecture and hardware-friendly implementation. According to different implementations of BMA, ME is categorized as unidirectional ME (UME) and bidirectional ME (BME) [9, 18, 19]. UME performs ME on the previous frame to generate MVs from the previous frame to the following frame, but it usually results in holes and overlapping. According to temporal symmetry, BME directly performs BMA on the interpolated frame and assigns a unique MV to each block, which avoids holes and overlaps, However, due to the unavailability of interpolated frames, BME often produces the inaccurate MVs, resulting in some blocking artifacts. To further improve the interpolation quality, Choi et al. [20] proposed a convolutional neural network (CNN) to predict the absent frames; Zhang et al. [21] proposed a deep residual network (DRN) to synthesize the interpolated results by weighting various predictions output by CNNs; and Khoubani and Moradi [22] proposed quaternion wavelet transform (QWT) to improve the ME accuracy. The above-mentioned methods can estimate the MVs more accurately, but they are not suitable for the hardware platform and real-time applications due to the heavy computational burden. Romano and Elad [23] use a self-similar descriptor [24] to represent the context features of each block, which effectively reduces the block mismatches. Motivated by Romano et al., we find the feature is helpful to suppress the inaccuracy of MVs in BME. However, we need a more effective feature to stand out the block characteristic, and the feature extraction cannot introduce excessive computations. Recently, many pretrained networks are used to extract the image features. Without the training stage, these pretrained networks can rapidly produce the features, and the extracted features are more effective than traditional ones due to the large-scale image data set being invested in advance. Therefore, it is necessary to explore how to fuse the pretrained network into MC-FRUC.

In this paper, we first extract the features of each video frame by the pretrained network; then, the extracted features are fused into the bidirectional matching to generate the MVs of the interpolated frame. According to the output MVs, the MCI is implemented to produce the interpolated frame. The main contributions of our work are described as follows:

(i) *Feature Extraction*. We use the pretrained network to extract the feature of each video frame. The pretrained network cannot introduce excessive computations, and extracted features are so rich as to improve the accuracy of BME.

(ii) *Feature Match*. In BME, the extracted features are combined with the video frame to perform a bidirectional match. To control the influence of extracted features, we also weigh the feature term in the matching cost function.

Experiment results show that the extracted feature effectively improves BME accuracy and provide good objective and subjective interpolation qualities.

The rest of this paper is organized as follows. The BME and pretrained networks are described in Section 2. The detailed processes of the proposed MC-FRUC algorithm are described in Section 3. Experimental results are shown in Section 4. Finally, we conclude this paper in Section 5.

2. Background

2.1. BME. To avoid holes and overlaps, most of FRUC methods use BME to produce the MVs of the interpolated frame. According to the assumption of temporal symmetry, each block in the interpolated frame is assigned to a unique MV. As shown in Figure 1, BME directly implements BMA on the intermediate frame $\mathbf{Y}_t$ to compute the MV of each block. BMA divides $\mathbf{Y}_t$ into non-overlapping blocks, and the MV of each block is estimated by analyzing the motion trajectories of the previous $\mathbf{Y}_{t-1}$ and next frame $\mathbf{Y}_{t+1}$. Let $\mathbf{B}_{i,j}$ denote the i-th row and j-th column block in $\mathbf{Y}_t$. The search window $\mathbf{W}_{i,j}$ in $\mathbf{Y}_{t-1}$ and $\mathbf{Y}_{t+1}$ is set to be $N \times N$ pixels in size. With any pixel in $\mathbf{W}_{i,j}$ as the center, the candidate matching blocks are be extracted, and each candidate block has a pair of symmetric MVs according to the assumption of temporal symmetry. In order to select the best MV from the set of candidate MVs, BME introduces the sum of bilateral absolute differences (SBAD) criterion. The SBAD of each candidate block is calculated, and the candidate block with the smallest SBAD value is located, and their relative displacement is computed from $B_{i,j}$ as the best MV, i.e.,

$$\begin{aligned} \mathbf{v}_{i,j} &= \arg\min_{\mathbf{v}} \left\{ \mathrm{SBAD}\left[\mathbf{B}_{i,j}, \mathbf{v}\right] \right\}, \\ \mathrm{SBAD}\left[\mathbf{B}_{i,j}, \mathbf{v}\right] &= \sum_{\mathbf{p} \in \mathbf{B}_{i,j}} \left| Y_{t-1}(\mathbf{p} + \mathbf{v}) - Y_{t+1}(\mathbf{p}\text{-}\mathbf{v}) \right|, \end{aligned} \tag{1}$$

where $Y_{t-1}(\mathbf{p})$ and $Y_{t+1}(\mathbf{p})$ represent the luminance values of the pixel $\mathbf{p}$ in $\mathbf{Y}_{t\text{-}1}$ and $\mathbf{Y}_{t+1}$, respectively; $\mathbf{p}$ denotes a pixel in $\mathbf{B}_{i,j}$; and v represent the MV of the candidate block.

Although BME avoids holes and overlaps in the interpolated frames, the true MV of the object does not always guarantee that the interpolated block has a minimum SBAD, especially for the occlusion and local similar area. To suppress the bad effects resulting from inaccuracy of MVs in BME, we propose that the features of each frame can be extracted using pretrained network. The following briefly introduces the pretrained network.

2.2. Pretrained Network. The pretrained network is a deep neural network that has already been trained on large data sets. It has two or more hidden layers, and these hidden layers include the convolutional layer, pooling layer, and the fully connected layer. There are many developed pretrained network, for example, AlexNet [25], VGG [26], ResNet [27],

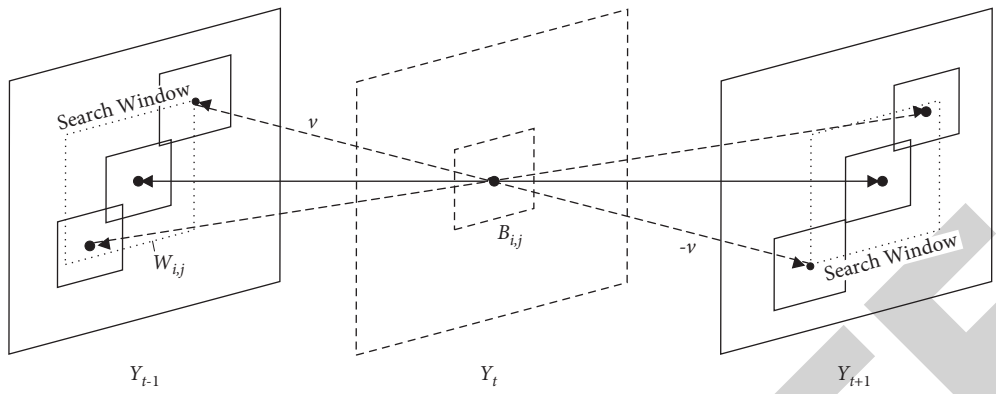


Figure 1: Mechanism of BME.

and so on, and they can be modified as the feature extractor. AlexNet is a network aiming at image classification, and it achieves excellent classification performance due to the effective extraction of the features of images. Figure 2 illustrates the structure of AlexNet. The first layer of AlexNet filters the $227 \times 227 \times 3$ input image in a stride of 4 by using 96 kernels of size $11 \times 11 \times 3$. The convolution layer is followed by a rectified linear unit (ReLU) and batch normalization (BN) transformation and the max pooling. The second layer takes the output of the first layer as the input and filters the input with 256 convolution kernels of size $5 \times 5 \times 48$. The ReLU and BN transformation are still performed, and the max-pooling operation is also added. In the third and fourth layer, ReLU is added after the convolution operation, and the convolution kernels are 3×3 in size. In the fifth layer, the max-pooling operation is performed in addition to implementing convolution and ReLU. In the last three layers, the full connection (FC) and ReLU are added, and the dropout is introduced to prevent overfitting. It generates a 1,000-dimensional feature vector by softmax in the output layer. From the above, it can be seen that AlexNet consists of five convolutional layers and three fully connected layers. It can effectively suppress the overfitting with the help of max pooling, and the range of values for the feature value can also be limited reasonably by using ReLU. AlexNet has achieved great success in the representation of the features, and it can output rich features. Therefore, we modify AlexNet as a feature extractor and fuse the extracted features into BMA to improve the BME accuracy.

3. Proposed MC-FRUC Algorithm

3.1. Framework Overview. Figure 3 presents the framework of the proposed MC-FRUC algorithm. First, the pretrained AlexNet is used to extract the previous frame $\mathbf{Y}_{t-1}$ and the following frame $\mathbf{Y}_{t+1}$ and produces the corresponding feature layers $\mathbf{F}_{t-1}$ and $\mathbf{F}_{t+1}$. The pretrained AlexNet cannot introduce excessive computations, and extracted features are so rich as to improve the accuracy of BMA. The sizes of the extracted $\mathbf{F}_{t-1}$ and $\mathbf{F}_{t+1}$ are the same as those of their corresponding $\mathbf{Y}_{t-1}$ and $\mathbf{Y}_{t+1}$, respectively. Then, $\mathbf{F}_{t-1}$ and $\mathbf{F}_{t+1}$ are combined with $\mathbf{Y}_{t\text{-}1}$ and $\mathbf{Y}_{t+1}$, respectively, to implement bidirectional match and generate motion vector field (MVF) $\mathbf{V}_t$. of the interpolation frame $\widehat{\mathbf{Y}}_t$. Finally, according to $\mathbf{V}_t$, the MCI is performed to generate the estimation $\widehat{\mathbf{Y}}_\mathbf{t}$ of $\mathbf{Y}_t$. The following describes the implementation of the MC-FRUC algorithm in detail.

3.2. Feature Extraction by Pretrained AlexNet. The pretrained network has the capability to extract the image feature by revising the network structure. In a pretrained network, the results output by each layer can be regarded as the feature. However, the higher the layer is, the richer the output features are. Therefore, we use the last layer of AlexNet as a feature extractor; the implementation is shown in Figure 4.

The improved AlexNet removes a layer of the fully connected layers, and the network model is divided into seven layers: the first five layers are convolution layers, and the next two layers are fully connected layers. First, each video frame is resized to the same size as the input layer in AlexNet. The input frame is filtered by a convolution kernel in Conv1. The ReLU and BN transformation is performed to improve the speed and accuracy of the training network, and a max-pooling operation is performed to enhance the richness of the feature. Then, all the convolution layers are traversed. Conv2 performs the convolution operation, ReLU and BN transformation, and max-pooling operation to get deeper features. Conv3 and Conv4 also perform the convolution operation and Conv5 implements max-pooling operation after the convolution operation. Finally, the fully connected layer connects the feature graph generated by Conv5 and produces a 4,096-dimensional feature vector in the Fc6 and Fc7. The features output by the Fc7 keep essential information of the input frame, and full description for the feature makes the video frame is more distinctive, so it benefits BMA to reduce block mismatches and improves the quality of interpolation frames.

Figure 5 presents the visualization of the extracted features by different layers of AlexNet. It can be seen that the different layers produce the features with different complexities. The extracted features by Conv 1 are shown in Figure 5(b). It can be seen that the features, highlight edges, brightness, and contrast. It can depict the texture of the character, but this layer extracts limited information. From

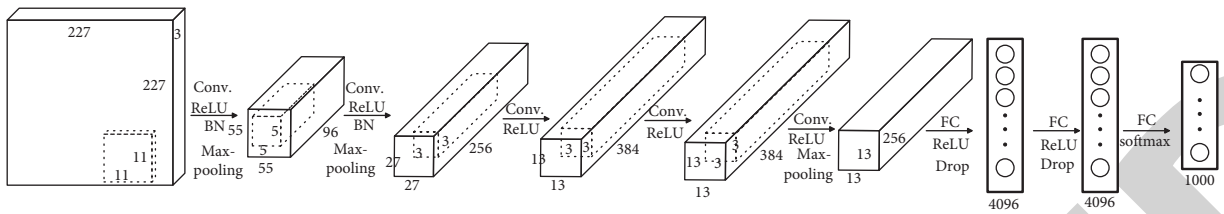


Figure 2: Structure of AlexNet.

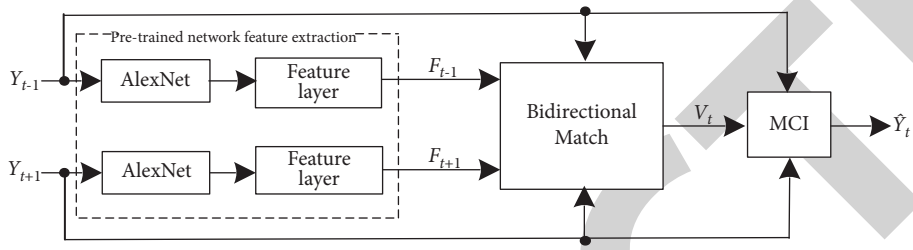


Figure 3: Framework of the proposed MC-FRUC algorithm.

Figure 5(c), the extracted features by Conv 2 enhance the textures and angles. Textures and edges are clearer than Conv 1. Figure 5(d) presents the features extracted from Conv3; we can see that Conv 3 produces richer features than the formers. The general outline of the figure is distinct. The features extracted by Conv4 and Conv5 are presented in Figures 5(e) and 5(f), respectively. It can be observed that extracted features become more and more concrete; for example, the features of the face are more obvious. Details are also extracted for the input frame. Furthermore, the features become rich. From Figures 5(g) and 5(h), we can see the output of the features by Fc6 and Fc7 that describe globally each video frame, and these features stand out some important local areas. All important features are extracted. The features can be integrated into BMA in BME to calculate the accurate motion vector and improve matching accuracy. Therefore, it can be found that the features of the fully connected layers are fused with BMA to improve the matching effect and the quality of interpolation frames. The following describes how to implement the bidirectional match based on the extracted features.

3.3. Bidirectional Match. The proposed bidirectional match fuses the extracted features into the BME framework. For the previous frame $\mathbf{Y}_{t-1}$ and the following frame $\mathbf{Y}_{t+1}$, the features extracted from pretrained AlexNet are combined as the feature layers $\mathbf{F}_{t-1}$ and $\mathbf{F}_{t+1}$. For i-th row and j-th column $\mathbf{B}_{i,j}$ in the interpolated frame $\mathbf{Y}_t$, we need to find its matching blocks in $\mathbf{Y}_{t-1}$ and $\mathbf{Y}_{t+1}$, so a search window $\mathbf{W}_{i,j}$ with the size of $N \times N$ is set in $\mathbf{Y}_{t-1}$ and $\mathbf{Y}_{t+1}$; all pixels in $\mathbf{W}_{i,j}$ are traversed to construct the candidate MV set $\Omega_{i,\,j}$. According to the assumption of temporal symmetry, for the candidate MV v in $\Omega_{i,\,j}$, we compute its matching cost as follows:

$$J\left[\mathbf{B}_{i,j}, \mathbf{v}\right] = \sum_{\mathbf{p} \in \mathbf{B}_{i,j}} \left\{ \left| Y_{t-1}(\mathbf{p} + \mathbf{v}) - Y_{t+1}(\mathbf{p} - \mathbf{v}) \right| + \beta \left| F_{t-1}(\mathbf{p} + \mathbf{v}) - F_{t+1}(\mathbf{p} - \mathbf{v}) \right| \right\}, \tag{2}$$

where $Y_{t-1}(\mathbf{p})$ and $Y_{t+1}(\mathbf{p})$ represent the luminance values of the pixel $\mathbf{p}$ in $\mathbf{Y}_{t-1}$ and $\mathbf{Y}_{t+1}$, respectively; $F_{t-1}(\mathbf{p})$ and $F_{t+1}(\mathbf{p})$ represent the values of the pixel $\mathbf{p}$ in $\mathbf{F}_{t-1}$ and $\mathbf{F}_{t+1}$, respectively; and β is the regularization factor to control the influence of extracted features. By comparing the matching costs of all candidate $\Omega_{i,\,j}$, the final MV $\mathbf{v}_{i,j}$ of $\mathbf{B}_{i,j}$ is determined by

$$\mathbf{v}_{i,j} = \arg\min_{v \in \Omega_{i,j}} \left\{ J\left[\mathbf{B}_{i,j}, \mathbf{v}\right] \right\}. \tag{3}$$

The bidirectional match takes into account pixel differences and their corresponding features differences, and it can effectively suppress occlusions and block mismatches. Therefore, BME accuracy is improved, leading to the enhancement of the interpolation quality.

4. Experimental Results

In this section, the performance of the proposed MC-FRUC algorithm is evaluated by transmitting the YUV sequences with a CIF format in a simulation environment of IoT. These sequences include *Foreman*, *Akiyo*, *Bus*, *Football*, *Mobile*, *Stefan*, *Tennis*, *Flower*, *News*, *City*, *Coastguard*, *Mother* & *Daughter*, and *Soccer*. The interpolated results by the proposed algorithm are compared with those that are generated by its two comparing algorithms proposed by Choi et al. [20] and Romano and Elad [23]. The comparing algorithms keep

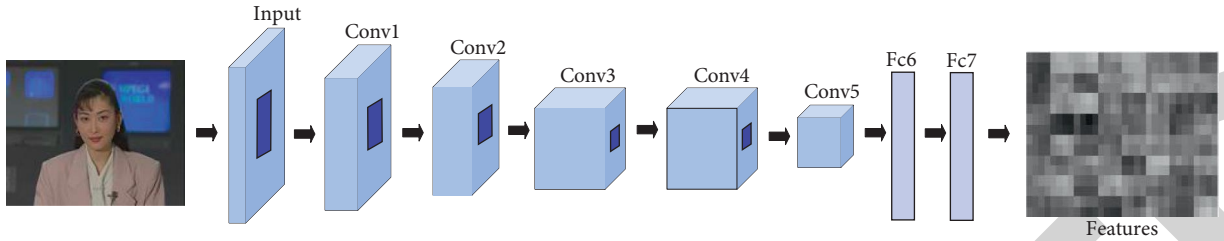


Figure 4: The flow chart of extracting features by AlexNet.

their original parameter settings except for the block size. In the proposed algorithm, the block size and the search window size are set to be 16 and 21, respectively. To evaluate the quality of the interpolated frames from subjective and objective perspectives, we transmit the odd frames of the video sequence from IoT nodes to the IoT cloud, and the cloud recovers the even frames according to the transmitted frames. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used to evaluate the differences between the restored frames and the original frames.

4.1. Objective Evaluation. Table 1 presents the average PSNR values, SSIM values, and execution time of test sequences recovered by Choi et al. [20], Romano and Elad [23], and the proposed algorithm. From Table 1, the proposed algorithm has higher PSNR values than Choi et al. [20] and Romano and Elad [23] in most cases. The average PSNR values of the proposed algorithm on all test sequences are 0.92 dB and 1.16 dB higher than those of Choi et al. [20] and Romano and Elad [23], respectively. Choi et al. [20] get a PSNR value 0.3 dB higher than the proposed algorithm on the *Akiyo* sequence, but the proposed algorithm has higher PSNR values than Choi et al. [20] and Romano and Elad [23] on other test sequences. Meanwhile, we can see that the proposed algorithm has obvious SSIM improvement over Choi et al. [20] and Romano and Elad [23]. Choi et al. [20] get SSIM value higher than Romano and Elad [23] and the proposed algorithm on the *Tennis* and *Soccer* sequences, but the proposed method has higher SSIM values than Choi et al. [20] and Romano and Elad [23] on other test sequences. These SSIM results indicate that the proposed algorithm can better retain the structural information of interpolated frames. For the execution time, it can be seen that the proposed algorithm costs the moderate execution time to interpolate a video frame, that is, Choi et al. [20] costs only 0.52 seconds to interpolate a frame on average. Romano and Elad [23] cost 13.12 s to interpolate a frame on average, and the proposed algorithm costs 2.03 seconds to interpolate a frame. The average PSNR gains of the proposed algorithm are higher than that of Choi et al. [20] and Romano and Elad [23] on all the test sequences under the same parameter setting, showing that the proposed MC-FRUC algorithm can generally provide better objective quality than those chosen comparing algorithms.

Figure 6 shows the PSNRs and SSIMs of individual interpolated frames on the *Foreman*, *Stefan*, *Mobile*, and *Bus* sequences. It can be seen that the PSNR and SSIM values of the most of recovered frames by the proposed algorithm are higher than the comparing algorithms. The performance of Choi et al. [20] and Romano and Elad [23] is the same, and they are both worse than the proposed algorithm. For *Mobile* and *Bus* sequences, Choi et al. [20] and Romano and Elad [23] are lower than the proposed algorithm, so the PSNR and SSIM curve of the proposed algorithm is close to the best one in the comparing algorithms. For *Foreman* and *Stefan* sequences, the proposed algorithm outperforms the comparing algorithms in most cases. And it is much higher than Choi et al. [20] and Romano and Elad [23]. From the above, it can be concluded that the proposed algorithm ensures better objective quality with moderate computational complexity, so the proposed algorithm is an effective way to improve interpolation quality.

4.2. Subjective Evaluation. Figure 7 presents the visual results on the 78th interpolated frame of the *Foreman* sequence using different FRUC algorithms. By comparing these results with the original frame, there are severe blurs in the nose and eyes region for the interpolated frames by Choi et al. [20] and Romano and Elad [23], and background boundary also produces ghost effects; however, the proposed algorithm provides a clear face and the unambiguous background boundary, producing the comfortable visual quality. Figure 8 presents the visual results on the 14th interpolated frame of *Stefan* sequence using different FRUC algorithms. For the results interpolated by Choi et al. [20] and Romano and Elad [23], the feet of sport man and the letters on the wall are recovered with annoying artifacts, but the proposed algorithm effectively suppresses these artifacts and presents better visual results. Figure 9 presents the visual results on the 50th interpolated frame of the *Mobile* sequence using different FRUC algorithms. The digital region of the calendar are disturbed in the interpolation results by Choi et al. [20] and Romano and Elad [23], and there are serious blurs over the rolling sphere and the train, but the proposed algorithm can clearly recover these numbers, and the blurs over the rolling sphere and the train are effectively suppressed. Figure 10 presents the visual results on the 62th interpolated frame of the *Bus* sequence using different FRUC algorithms. For the interpolated results by Choi et al. [20] and Romano et al. [23], the front of the Bus is recovered unclearly, and the iron fences are also misplaced, but the proposed algorithm produces the satisfying visual quality. From the above results, it can be seen that the proposed algorithm can provide good subjective quality.

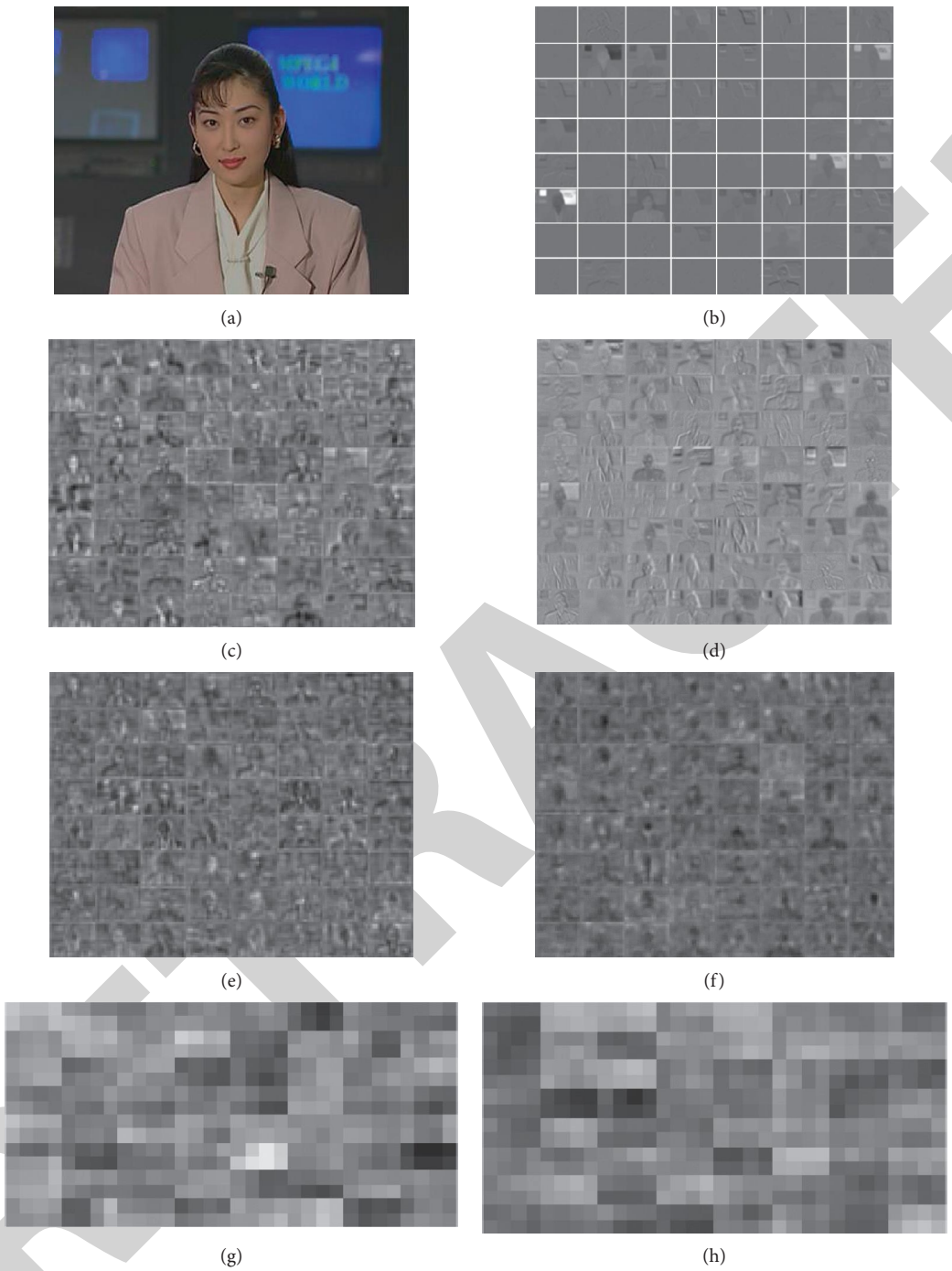
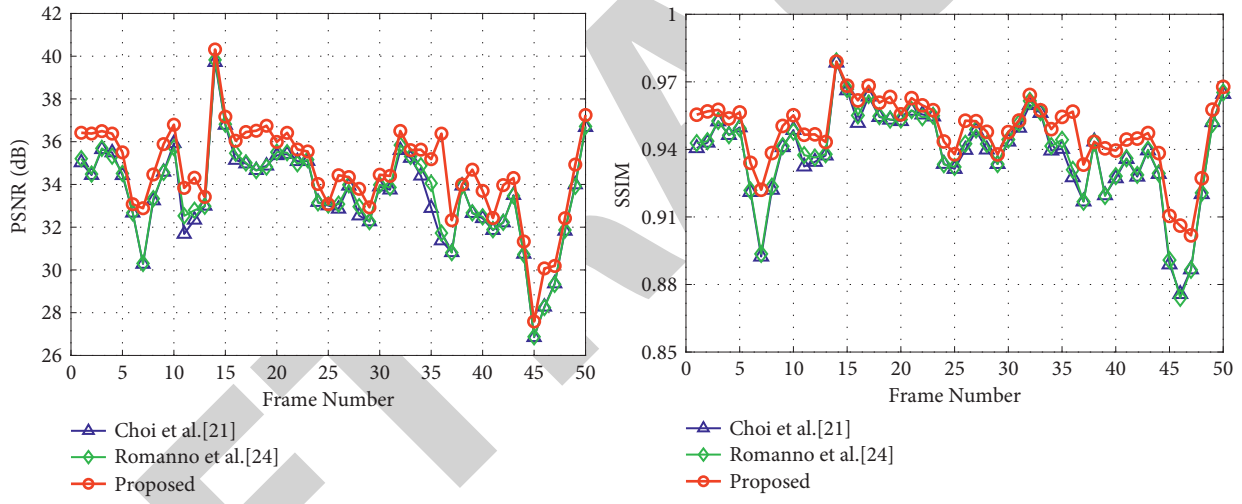


Figure 5: Extracted features: (a) input, (b) visualization of feature in Conv1, (c) visualization of feature in Conv2, (d) visualization of feature in Conv3, (e) visualization of feature in Conv4, (f) visualization of feature in Conv5, (g) visualization of feature in Fc6, and (h) visualization of feature in Fc7.

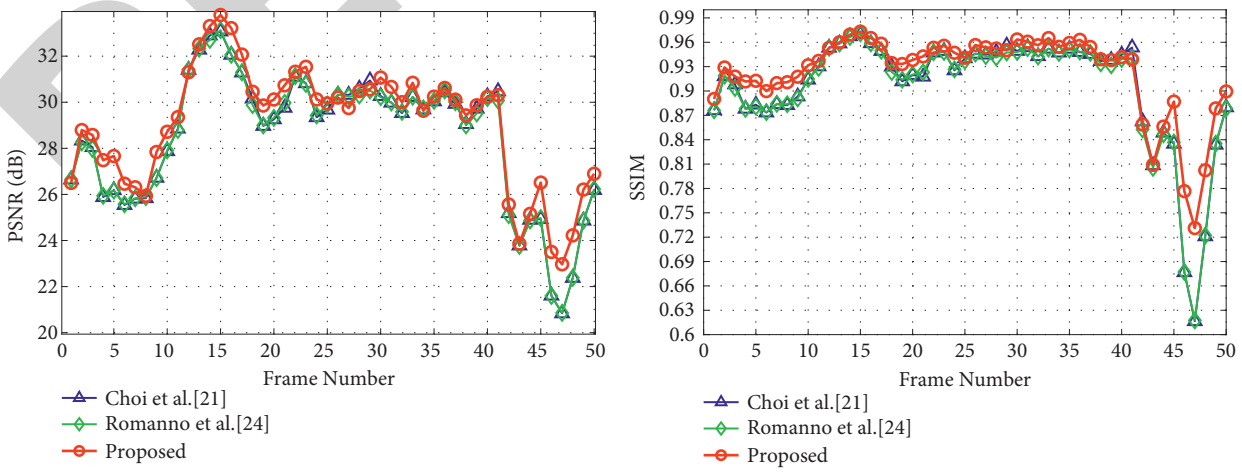
Table 1: Average PSNR (dB), SSIM, and execution time (s/frame) comparisons of test CIF sequences recovered by Choi et al. [20], Romano and Elad [23], and the proposed algorithm.

Sequence	Choi et al. [20]			Romano and Elad [23]			Proposed		
	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
Foreman	33.52	0.9384	0.57	33.62	0.9392	13.49	**34.65**	0.9482	2.01
Akiyo	**46.37**	0.9956	0.53	43.40	0.9843	13.39	46.07	0.9942	2.03
Bus	26.26	0.8984	0.51	26.26	0.8985	13.56	**27.45**	0.9222	2.00
Football	22.69	0.6520	0.53	22.61	0.6493	13.42	**22.95**	0.6688	2.02
Mobile	25.23	0.8787	0.53	25.29	0.8803	13.47	**28.82**	0.9507	2.00
Stefan	28.49	0.9056	0.53	28.46	0.9042	13.55	**29.06**	0.9228	2.00
Tennis	29.39	**0.8901**	0.53	29.06	0.8775	13.13	**29.96**	0.8812	2.00
Flower	30.12	0.9623	0.53	30.19	0.9641	13.31	**30.76**	0.9729	2.03
News	36.54	0.9781	0.50	36.47	0.9778	12.68	**37.40**	0.9809	2.03
City	32.30	0.9180	0.50	32.09	0.9130	12.70	**33.23**	0.9300	2.08
Coastguard	32.21	0.9131	0.51	32.21	0.9131	12.62	**32.63**	0.9212	2.07
Mother & Daughter	41.10	0.9767	0.51	41.64	0.9770	12.55	**42.46**	0.9789	2.07
Soccer	28.32	**0.9019**	0.51	28.07	0.8861	12.67	**29.01**	0.8953	2.09
Avg.	31.73	0.9084	0.52	31.49	0.9050	13.12	**32.65**	0.9206	2.03

The bold values represent the optimal values in the comparison of test results.



(a)



(b)

Figure 6: Continued.

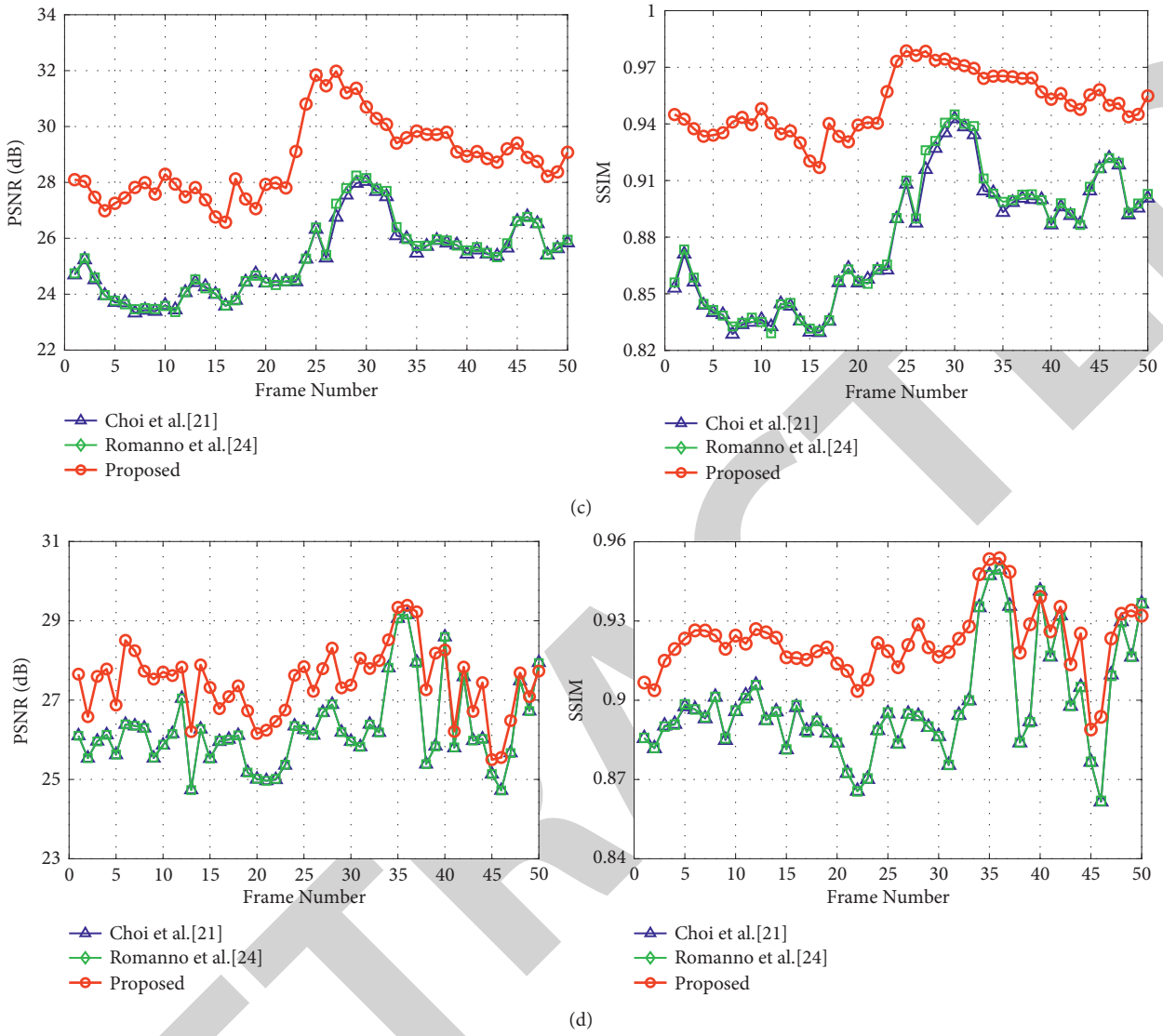
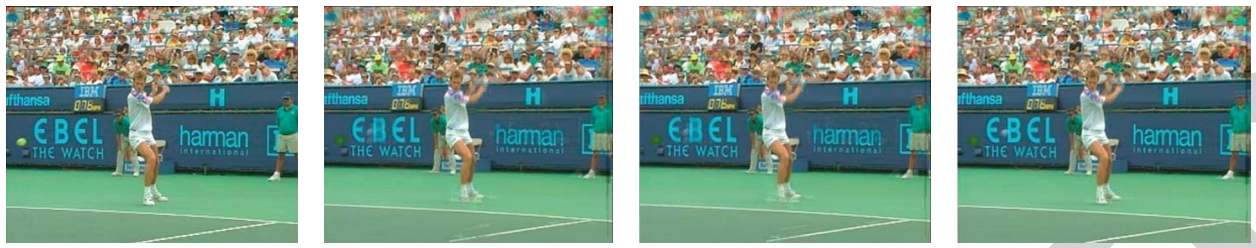


Figure 6: PSNRs and SSIMs of the interpolated frames constructed by Choi et al. [20], Romano and Elad [23], and the proposed algorithm for: (a) *Foreman*, (b) *Stefan*, (c) *Mobile*, and (d) *Bus*. In each subfigure, the evaluation criteria are PSNR and SSIM from left to right.

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

Figure 7: Visual results on the 78th interpolated frame of *Foreman* sequence with different FRUC algorithms: (a) original, (b) Choi et al., (c) [20] Romano and Elad [23], and (d) proposed.



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

Figure 8: Visual results on the 14th interpolated frame of the *Stefan* sequence using different FRUC algorithms: (a) original, (b) Choi et al. [20], (c) Romano and Elad [23], and (d) proposed.

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

Figure 9: Visual results on the 50th interpolated frame of the *Mobile* sequence using different FRUC algorithms: (a) original, (b) Choi et al. [20], (c) Romano and Elad [23], and (d) proposed.

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

Figure 10: Visual results on the 62th interpolated frame of *Bus* sequence using different FRUC algorithms: (a) original, (b) Choi et al. [20], (c) Romano and Elad [23], and (d) proposed.

5. Conclusions

In this paper, the pretrained AlexNet is used to design an MC-FRUC algorithm, which is applied to video communication in IoT. First, the pretrained AlexNet is constructed, and its output of the fully connected layer is used as the features of each video frame. Second, the extracted features are fused into the BME framework to produce the MVF of the interpolated frame and suppress the block mismatches and occlusions. Finally, according to the output MVF, the MCI is performed to interpolate the absent frame. The performance of the proposed algorithm is evaluated by testing video sequences in the simulation environment of IoT. Experimental results show that the proposed MC-FRUC algorithm can improve the BME accuracy, and achieve better objective and subjective qualities.

In future work, we will focus on the development of new efficient ways for more accurate ME. Furthermore, how to improve the quality of video communication in IoT is worthy of investigation. We plan to extend our analysis by considering more powerful deep learning methods.

Data Availability

The experimental codes have been downloaded from Ran Li's homepage: http://www.scholat.com/liran358

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was funded in part by the Project of Science and Technology Department of Henan Province in China, under Grant no. 212102210106; in part by the National Natural Science Foundation of China, under Grant no. 31872704; in part by Innovation Team Support Plan of University of Science and Technology of Henan Province, under Grant no. 19IRTSTHN014; and in part by the Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing and China Ministry of Education Key Laboratory of Cognitive Radio and Information Processing and supported by the Scientific Research Foundation of Graduate School of Xinyang Normal University, under Grant no. 2020KYJJ39.

References

[1] T.-H. Hsu and Y.-M. Tung, "A social-aware P2P video transmission strategy for multimedia IoT devices," *IEEE Access*, vol. 8, Article ID 95574, 2020.

[2] K. Muhammad, T. Hussain, M. Tanveer, G. Sannino, and V. H. C. de Albuquerque, "Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4455–4463, 2020.

[3] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: multiaccess edge computing for 5G and internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.

[4] D. Vranjes, S. Rimac-Drlje, and M. Vranjes, "Adaptive temporal frame interpolation algorithm for frame rate up-conversion," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 17–21, 2020.

[5] R. Vanam and Y. A. Reznik, "Frame rate up-conversion using bi-directional optical flows with dual regularization," in *Proceedings of the 2020 IEEE International Conference on Image Processing*, pp. 558–562, Abu Dhabi, UAE, October 2020.

[6] 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. E94-C, no. 5, pp. 896–904, 2011.

[7] N. Van Thang, K. Lee, and H.-J. Lee, "A Stacked deep MEMC network for frame rate up conversion and its application to HEVC," *IEEE Access*, vol. 8, Article ID 58310, 2020.

[8] W. Song, P. Heo, G. Choi, S. R. Oh, and H. Park, "Motion compensated frame interpolation of occlusion and motion ambiguity regions using color-plus-depth information," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, October. 2018.

[9] G. Chen, "Frame rate up-conversion algorithm based on adaptive-agent motion compensation combined with semantic feature analysis," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 511–518, 2018.

[10] 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, Article ID 60353, 2018.

[11] L. Zhou, R. Sun, X. Tian, and Y. Chen, "Phase-based frame rate up-conversion for depth video," *Journal of Electronic Imaging*, vol. 27, 2018.

[12] X. Song, J. Yao, L. Zhou et al., "A practical convolutional neural network as loop filter for intra frame," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing*, Athens, Greece, October 2018.

[13] J. Panda and S. Meher, "An efficient image interpolation using edge-error based sharpening," in *Proceedings of the 2020 IEEE 17th India Council International Conference (INDICON)*, pp. 1–6, New Delhi, India, December 2020.

[14] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 407–416, 2007.

[15] S. Seong-Gyun Jeong, C. Chang-Su Kim, and C. Kim, "Motion-compensated frame interpolation based on m motion estimation and texture optimization," *IEEE Transactions on Image Processing*, vol. 22, no. 11, pp. 4497–4509, 2013.

[16] D. Kim, H. Lim, and H. Park, "Iterative true motion estimation for motion-compensated frame interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 3, pp. 445–454, 2013.

[17] Q. Lu, N. Xu, and X. Fang, "Motion-compensated frame interpolation with m-based occlusion handling," *Journal of Display Technology*, vol. 12, no. 1, pp. 45–54, 2016.

[18] H. N. Mirarkolaei, S. R. Snare, A. H. Schistad Solberg, and E. N. Steen, "Frame rate up-conversion in cardiac ultrasound," *Biomedical Signal Processing and Control*, vol. 58, Article ID 101863, 2020.

[19] G. Choi, P. Heo, S. R. Oh, and H. Park, "A New motion estimation method for motion-compensated frame interpolation using a convolutional neural network," in *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 800–804, Beijing, China, September 2017.

[20] G. Choi, P. Heo, and H. Park, "Triple-frame-based Bi-directional motion estimation or motion-compensated frame interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1251–1258, 2019.

[21] Y. Zhang, L. Chen, C. Yan, P. Qin, X. Ji, and Q. Dai, "Weighted convolutional motion-compensated frame rate up-conversion using deep residual network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 11–22, 2020.

[22] S. Khoubani and M. H. Moradi, "A fast qwbmcfrucwfsatete," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8999–9025, 2021.

[23] Y. Romano and M. Elad, "Con-patch: when a patch meets its context," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 3967–3978, 2016.

[24] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Lake Tahoe, USA, December 2012.

[26] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.