A Deep Learning Approach for Multi-Frame In-Loop Filter of HEVC

Tianyi Li, Mai Xu, *Senior Member, IEEE*, Ce Zhu, *Fellow, IEEE*, Ren Yang, Zulin Wang, and Zhenyu Guan

***Abstract*—An extensive study on the in-loop filter has been proposed for a high efficiency video coding (HEVC) standard to reduce compression artifacts, thus improving coding efficiency. However, in the existing approaches, the in-loop filter is always applied to each single frame, without exploiting the content correlation among multiple frames. In this paper, we propose a multi-frame in-loop filter (MIF) for HEVC, which enhances the visual quality of each encoded frame by leveraging its adjacent frames. Specifically, we first construct a large-scale database containing encoded frames and their corresponding raw frames of a variety of content, which can be used to learn the in-loop filter in HEVC. Furthermore, we find that there usually exist a number of reference frames of higher quality and of similar content for an encoded frame. Accordingly, a reference frame selector (RFS) is designed to identify these frames. Then, a deep neural network for MIF (known as MIF-Net) is developed to enhance the quality of each encoded frame by utilizing the spatial information of this frame and the temporal information of its neighboring higher-quality frames. The MIF-Net is built on the recently developed DenseNet, benefiting from its improved generalization capacity and computational efficiency. In addition, a novel block-adaptive convolutional layer is designed and applied in the MIF-Net, for handling the artifacts influenced by coding tree unit (CTU) structure in HEVC. Extensive experiments show that our MIF approach achieves on average 11.621% saving of the Bjøntegaard delta bit-rate (BD-BR) on the standard test set, significantly outperforming the standard in-loop filter in HEVC and other state-of-the-art approaches.**

***Index Terms*—High efficiency video coding, in-loop filter, deep learning, multiple frames.**

I. Introduction

Recently, the rapid growth of high definition videos has brought about increasing visual experience, but meanwhile posing the challenge on transmitting or storing the huge amount of video data. Addressing the challenge, the Joint Collaborate Team on Video Coding (JCT-VC) has proposed the high efficiency video coding (HEVC) standard [1] for video compression. Compared with its predecessor H.264/advanced video coding (AVC) standard [2], HEVC can save approximately 50% bit-rate on average. This benefits from an integration of advanced coding techniques, e.g., the flexible quad-tree-based structure of coding tree unit (CTU), the increased number of intra-prediction modes and the more precise interpolation for motion compensation. However, various compression artifacts (e.g., blocking, blurring and ringing artifacts) [3] still present in compressed videos, especially at low bit-rates. The artifacts mainly result from the block-wise prediction and quantization with limited precision. To alleviate the compression artifacts, in-loop filters were adopted as crucial components in the recent video coding standards, via enhancing the quality of each encoded frame and providing higher-quality reference for its successive frames. Consequently, the coding efficiency can be further improved by adopting the in-loop filters.

In total, three types of built-in in-loop filters were proposed for the standard HEVC, including deblocking filter (DBF) [4], sample adaptive offset (SAO) filter [5] and adaptive loop filter (ALF) [6]. These in-loop filters are implemented sequentially in the HEVC. Specifically, DBF is firstly used to remove the blocking artifacts. Then the SAO filter reduces sample distortion by adding an adaptive offset to each sample. In addition, ALF was also considered to be implemented after the SAO filter, which can further minimize the mean square error between the reconstructed frames and the raw frames based on Wiener filter. However, ALF cannot provide visually better quality, and thus it was not adopted in the final version of HEVC. Besides the built-in in-loop filters for the HEVC, some other in-loop filtering methods were also proposed, containing both heuristic and learning-based methods. In heuristic methods [7]–[10], some prior knowledge of video coding is utilized to build a statistical model of compression artifacts, and then a filtering process is derived for enhancing quality of each video frame based on the model. In the most recent years, deep learning has been successfully employed in many areas about data compression, such as video coding [11], quality enhancement [12] and feature encoding [13], [14]. Also, learning-based methods have successfully enhanced the performance of in-loop filtering [15]–[20]. These methods typically adopt convolutional neural networks (CNN) to learn the spatial correlation of content within a frame patch. For example, Dai *et al.* [16] introduced a variable-filter-size residue-learning CNN (VRCNN) in place of the standard DBF

Manuscript received December 19, 2018; revised April 26, 2019; accepted May 29, 2019. Date of publication June 14, 2019; date of current version August 28, 2019. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61876013 and Grant 61573037, and in part by the Fok Ying Tung Education Foundation under Grant 151061. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jiwen Lu. *(Corresponding author: Mai Xu.)*

T. Li, R. Yang, and Z. Wang are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China.

M. Xu is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, and also with the Hangzhou Innovation Institute, Beihang University, Beijing 100191, China (e-mail: maixu@buaa.edu.cn).

C. Zhu is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China.

Z. Guan is with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China.

Digital Object Identifier 10.1109/TIP.2019.2921877

and SAO at intra-mode. Compared with the single-path CNN, variable filter sizes in [16] enable feature extraction at different spatial scales, with less network complexity and accelerated training process. Recently, Zhang *et al.* [20] have proposed a residual highway CNN (RHCNN) suitable for both intra- and inter-mode video coding, which is employed after the standard SAO. However, none of the above learning-based methods has employed multiple adjacent frames for in-loop filtering in the HEVC. As measured in this paper, high fluctuation of visual quality exists across the encoded frames in the HEVC, and thus a low-quality frame can be enhanced by referring to its adjacent higher-quality frames. Thus, it is possible to further reduce the compression artifacts of each encoded frame in the in-loop filters by using its adjacent frames, inspired by the works in multi-frame super-resolution [21]–[27].

Based on deep learning, this paper develops a multi-frame in-loop filter (MIF) for HEVC, replacing the original DBF and SAO. Specifically, we first construct a large-scale database for in-loop filtering in HEVC.[1] Our database contains distorted frames and their corresponding raw frames, generated from 182 raw video sequences at four quantization parameter (QP) values. Next, we need to examine the quality fluctuation of encoded frames in the HEVC. To this end, we design a reference frame selector (RFS) to search for higher-quality reference frames given an unfiltered reconstructed frame (URF), which is based on frame quality and content similarity. If RFS provides sufficient reference frames, the URF flows through a deep neural network for MIF (named MIF-Net) to utilize both spatial information within one frame and temporal information across frames. In MIF-Net, the content of each reference frame is first aligned with the URF via motion compensation, and then the URF is enhanced leveraging the information from multiple frames. In the case that no sufficient reference frame is selected by RFS, a simpler deep neural network for in-loop filter (named IF-Net) is used to enhance the URF instead. Both the MIF-Net and IF-Net are built upon the recently developed DenseNet [28], benefiting from its great success on the improved generalization capacity and computational efficiency. Also, considering the blocking artifacts highly influenced by the CTU partition structure, the proposed networks are also adaptive to the coding unit (CU) and transform unit (TU) partition in the HEVC, via varying convolutional kernels at different locations of the CU and TU grids. Finally, a mode selection scheme and the corresponding syntax are also designed to select the best mode among the three possible choices (i.e., MIF-Net, IF-Net and the standard in-loop filters), ensuring the overall performance of our approach. Figure 1 shows an example of our MIF approach. Here, the current 182nd frame is encoded as a URF, and then the 177th and 178th encoded frames are selected as its reference, with higher quality and similar content. Then, the URF and two reference frames are input to MIF-Net. As a result, the content with conspicuous artifacts (face and ear behind the bubble) in the URF can be significantly enhanced by MIF-Net, leveraging the information of two reference frames.

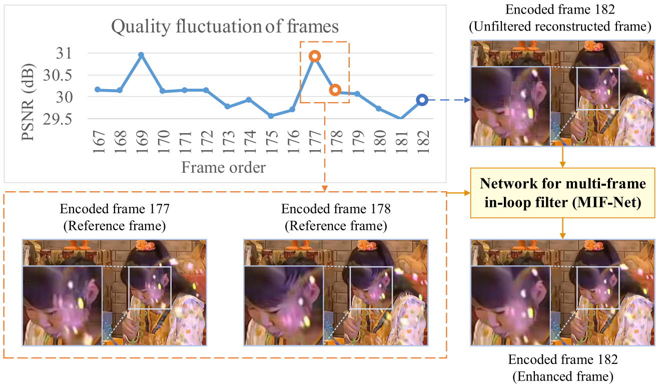


Fig. 1. An example illustrating quality fluctuation of frames and the proposed MIF approach.

This paper was previously presented in Data Compression Conference 2019 [29], with the following improvements. First, the training and validation sets of HIF database are enlarged from 93 sequences in [29] to 160 sequences. In addition, this paper thoroughly analyzes both frame quality and content similarity of compressed HEVC, as the foundation of our MIF approach. Next, we advance our MIF approach by developing the syntax regulation. Finally, we provide more extensive experimental results with various settings and the ablation study, verifying the effectiveness and the generalization ability of our MIF approach. In brief, the main contributions of this paper are summarized below.

- We construct a large-scale database for learning the in-loop filter of HEVC, with the potential to facilitate the further research in designing in-loop filters for HEVC encoding.
- We investigate the quality fluctuation of encoded frames in HEVC, and design an RFS to find higher-quality reference frames for URFs.
- We propose an MIF-Net and an IF-Net to prominently enhance the frame quality via utilizing both spatial and temporal information.

The rest of this paper is organized as follows. Section II reviews the related works on HEVC in-loop filtering and multi-frame super-resolution. Section III presents the constructed HIF database and examines the frame quality fluctuation in the HEVC. In Section IV, we propose the MIF approach for in-loop filtering, and Section V regulates the corresponding syntax. Section VI reports the experimental results to verify the effectiveness of the proposed approach. Finally, Section VII concludes this paper.

II. Related Works

In recent years, many in-loop filtering approaches have been proposed to improve the coding efficiency of HEVC by reducing the compression artifacts. Along with the development of HEVC, three built-in in-loop filters were designed, including DBF [4], SAO filter [5] and ALF [6]. Specifically, DBF, simplified from that in H.264, is adopted as the first in-loop filter of HEVC to remove the blocking artifacts at prediction unit (PU) or TU boundaries. Afterwards, the SAO filter refines samples in both smooth and textured areas. To this end, the SAO filter divides the samples into different

[1] Available at: https://github.com/tianyili2017/HIF-Database

categories and then adds an offset to each sample according to the category. In addition, ALF was also considered during the development of HEVC, which estimates suitable filter coefficients using Wiener filter at the encoder-end and then signals the coefficients to the decoder-end. However, it was not adopted in HEVC eventually, since it is unable to produce visually better quality.

In addition to the above built-in filters of HEVC, some other in-loop filtering methods have also been proposed. These methods can be classified into two categories, i.e, heuristic and learning-based methods. In heuristic methods [7]–[10], the statistical characteristics of artifacts are modeled according to some prior knowledge (such as textural complexity, and the number of similar frame patches), and a filtering process is then derived based on the model. For example, Matsumura *et al*. [7] introduced a non-local means (NLM) filter to HEVC, which takes the weighted mean of non-local similar frame patches for artifact reduction. The non-local design compensates the disadvantage that the pre-existing in-loop filters utilize only local information of frames. Ma *et al*. [9] developed a group-based in-loop filter to exploit both local and non-local similarities. With the obtained similarities, a reconstructed frame is firstly divided into multiple patch groups, and each group forms a matrix. Then, a soft or hard thresholding is applied to the singular values of the formed matrix, for achieving a sparse representation and meanwhile filtering out compression artifacts. Also based on the singular value decomposition of the group matrix, Zhang *et al.* [10] formulated the in-loop filtering as an optimization problem with low-rank constraint on every patch group, and then established an adaptive soft-thresholding model for sparse representation. Although the above heuristic methods have considerably enhanced the coding efficiency, the prior knowledge in these methods need to be manually exploited. Thus, the handcraft feature extraction results in inefficiency to some extent for the above heuristic methods. Meanwhile, it is also intractable to build a multi-variable filtering model, thus leading to limited coding efficiency enhancement in the above methods.

More recently, a number of learning-based in-loop filters have been proposed for HEVC, to address the shortcomings of heuristic methods. The learning-based methods can automatically learn the extensive features of compression artifacts and optimize the in-loop filters with sufficient trainable parameters. Since the input to the in-loop filter is always a frame patch of two dimensions, these methods typically adopt CNN to learn the spatial correlation of patch content. Specifically, Park *et al*. [15] utilized a four-layer super-resolution CNN (SRCNN) [30] to replace SAO in the encoding process. Dai *et al.* [16] introduced a VRCNN in place of DBF and SAO. Compared with a single-path CNN, variable filter sizes in [16] are helpful to extract features in different spatial scales, with less network complexity and accelerated training process. As we have witnessed tremendous progresses in CNN, some new CNN structures were also applied in in-loop filtering. For example, Kang *et al.* [17] proposed a multi-modal/multi-scale CNN to replace the existing DBF and SAO at intra-mode. This architecture mainly contains two convolutional sub-networks with different scales, also exploiting the CU and TU boundaries as input. Meng *et al.* [19] developed a multi-channel long-short-term dependency residual network (MLS-DRN) for mapping a distorted frame to its associated raw frame, inserted between DBF and SAO. Zhang *et al.* [20] investigated the performance of residual units with various internal structures, and proposed an RHCNN to build accurate mappings between the reconstructed frames and their corresponding raw frames. The RHCNN is employed as a high-dimensional filter after SAO, without conflicting the present in-loop filters.

It is also worth mentioning that the versatile video coding (VVC) standard is being developed by the Joint Video Exploration Team (JVET), as the successor to HEVC. Some proposals for JVET have already investigated deep-learning-based in-loop filters for VVC, containing two main categories, i.e., sequence-dependent and sequence-independent approaches. In the sequence-dependent approaches, a deep neural network model is trained on-line for certain frames and then used for in-loop filter for all frames of the same video sequence. For example, Hsiao *et al.* [31] proposed packing co-located luma and chroma patches together and then processing them with a three-layer CNN, for predicting the enhanced patches. The CNN is only trained on frames with temporal ID of 0 or 1, for reducing the computation overhead. Yin *et al.* [32] proposed training up to six CNN models on 8 encoded frames in each random access segment. As such, the best CNN model can be selected for each CTU of all encoded frames. These sequence-dependent approaches can learn a model adaptive to the specific content in a video sequence, while suffering a shortcoming that the on-line training inevitably introduces computation overhead. Instead, sequence-independent approaches have been more widely studied, in which a network model is trained offline and used for any video sequence. Specifically, Kawamura *et al.* [33] proposed a four-layer CNN with 3 × 3 taps, adopted after the DBF of VVC. Lin *et al.* [34] took QP into account and designed a deeper CNN, replacing the DBF and the SAO filter at intra-mode. Exploiting more advanced network typologies, Dai *et al.* [35] leveraged residue learning technique into a deep CNN, and meanwhile some trainable parameters are shared to save memory usage and prevent over-fitting. In addition, Wang *et al.* [36] developed a dense residual CNN based in-loop filter, with more flexible pathways across layers.

However, to the best of our knowledge, no existing work has employed multiple adjacent frames for in-loop filtering in the HEVC encoder. In this paper, we find it possible to further reduce compression artifacts of each encoded frame by using its adjacent frames, which is inspired by the existing works in multi-frame quality enhancement and super-resolution presented as follow. Most recently, Yang *et al.* [37] have proposed a decoder-end quality enhancement approach for HEVC. In [37], a support vector machine (SVM) based peak quality frame (PQF) detector first distinguishes PQFs from others, and then a novel CNN structure is applied to enhance each non-PQF according to its adjacent PQFs after motion compensation. In addition to quality enhancement, more works have been proposed for multi-frame super-resolution. In early years, some traditional signal processing and machine learning methods were proposed in [21], [22] for multi-frame super-resolution, increasing video resolution with reference to high-resolution key frames. Afterwards, deep learning was widely employed in this area. For example, Kappeler *et al.* [23] developed a video super-resolution network (VSRnet), where consecutive frames are firstly aligned together via motion compensation and then fed into a CNN that outputs super-resolved frames. Later, Li and Wang [24] proposed replacing the VSRnet by a deeper network based on residual learning. Recently, Huang *et al.* [25] proposed a bi-directional recurrent convolutional network (BRCN) for efficient multi-frame super-resolution, achieving both better performance and faster speed. Besides, [26] and [27] also presented other deep-learning-based super-resolution approaches for videos.

The above super-resolution methods [21]–[23], [25]–[27] and the decoder-end quality enhancement approach [37] are built on the assumption that, the same objects or scenes may appear in several successive frames and thus the content in a low resolution/quality frame can be inferred from its adjacent higher resolution/quality frames. Accordingly we make the first attempt to apply MIF at the encoder-end, which, we infer, has a great potential to improve coding efficiency.

III. Database for HEVC in-Loop Filter

A. Database Construction

We construct a large-scale database for HEVC in-loop filter (known as HIF database), to provide sufficient training data for the proposed approach and facilitate the subsequent works. For constructing the HIF database, 182 raw video sequences were collected, consisting of 6 sequences from [38], 87 sequences from Xiph.org [39] and 89 sequences from the Consumer Digital Video Library [40] in the Video Quality Experts Group (VQEG) [41]. These 182 sequences can be freely used for research without any commercial purpose. Note that 18 sequences of Classes A $\sim$ E from the Joint Collaborative Team on Video Coding (JCT-VC) test set [42] are also used for evaluating our MIF approach. However, these JCT-VC sequences are protected by copyrights, thus not included in our HIF database. Despite that, our database contains 182 downloadable video sequences, sufficient for training a deep-learning-based in-loop filter. The details about the sequences are listed in Table I. Considering that only resolutions in multiples of the minimum CU size (8×8 by default) are supported in HM [43], the NTSC sequences were cropped to 720×480 by removing the bottom edges of the frames. Moreover, the sequences longer than 10 seconds were clipped to be 10 seconds, preventing the over-large video files in our database.

All the sequences in our HIF database were divided into non-overlapping sets of training (120 sequences), validation (40 sequences) and test (22 sequences). Note that the 22 test sequences were randomly selected from [38], [39], [41] with five different resolutions and diverse content. The sequences were all encoded by HM 16.5 [43] at four QPs $\{22, 27, 32, 37\}$[2] with the Low Delay P (LDP) (using *encoder_lowdelay_P_main.cfg*), the Low Delay B (LDB) (using *encoder_lowdelay_main.cfg*) and the Random Access (RA) (using *encoder_randomaccess_main.cfg*) configurations. During the encoding procedure, all URFs (i.e., the reconstructed frames before DBF and SAO) were extracted as the input to MIF-Net, with their corresponding raw frames being ground-truth. In addition, CU and TU partition results for all the frames were also extracted as auxiliary features, since the compression artifacts are highly influenced by the block partition structure in HEVC. As a result, each frame-wise sample in the HIF database consists of four parts, i.e., a URF, its associated raw frame and two matrices indicating the CU and TU depth throughout the frame. Finally, 12 sub-databases were obtained corresponding to different QPs and configurations. As indicated in Table I, each sub-database contains 51,335 frames, and thus 616,020 frame-wise samples were collected for the whole HIF database. Note that each frame-wise sample can be split into multiple block-wise samples for data augmentation. Also, the position of each block-wise sample within the frame-wise sample is alterable, further increasing the variety of training samples in practice. Therefore, the HIF database is ready for providing sufficient data for our deep-learning-based MIF.

TABLE I
Sequences in HIF Database

Source	Resolution	Num. of sequences	Total num. of frames
Facial videos [38]	1920×1080 (1080p)	6	1,200
VQEG [41]	1920×1080 (1080p)	30	10,253
	640×360 (360p)	59	8,040
Xiph.org [39]	2048×1080 (2K)	18	9,248
	1920×1080 (1080p)	24	7,757
	1280×720 (720p)	4	2,100
	704×576 (4CIF)	5	2,880
	720×486 (NTSC)	7	2,100
	352×288 (CIF)	25	7,080
	352×240 (SIF)	4	677
Aggregated		182	51,335

B. Data Analysis

In this section, we analyze the quality fluctuation and content similarity of encoded frames, which serves as a premise for the proposed MIF. For such analysis, the default settings of the LDP, LDB and RA configurations are used, where the hierarchical coding of frames with periodical quality fluctuation is an inherent feature. However, as far as we know, the periodical quality fluctuation has not been quantified for designing the in-loop filters of HEVC. The purpose of this section is to quantify the frame quality and content similarity of compressed frames. Firstly, the quality of video frames is measured by peak signal to noise rate (PSNR), and the standard deviation (STD) of PSNR is used to evaluate the quality fluctuation. Figure 2 shows the fluctuation of PSNR in two selected sequences as examples, encoded at various configurations. It can be observed that evident fluctuation always presents across the frames in the same video sequence. For overall analysis, we further calculate the STD of frame-level PSNR for each sequence, as shown in Table II. Also, the PSNR increase by the original DBF and SAO in HEVC (named IDS in Table II) is

[2] In this paper, each mentioned QP represents the QP configured before encoding (equal to the QP of the first I-frame in a sequence), despite the QP may fluctuate during encoding.

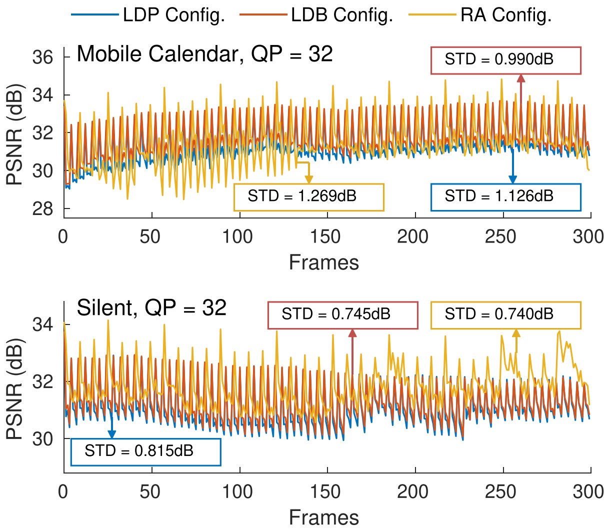


Fig. 2. PSNR fluctuation of two encoded sequences under various configurations.

TABLE II

STD AND IDS OF PSNR (DB) FOR TRAINING AND VALIDATION SEQUENCES

Config.	Metric	QP				Average
		22	27	32	37	
LDP	STD	0.855	0.910	0.923	0.876	0.891
	IDS	0.073	0.068	0.062	0.053	0.064
LDB	STD	0.827	0.897	0.923	0.880	0.882
	IDS	0.046	0.049	0.050	0.046	0.048
RA	STD	1.012	0.942	0.907	0.853	0.929
	IDS	0.047	0.051	0.053	0.050	0.050

provided for comparison. Note that the STD and IDS of PSNR are averaged on all the 160 training and validation sequences. From this table, we can find that the average STD of frame quality is 0.891 dB, 0.882 dB and 0.929 dB under the LDP, LDB and RA configurations, respectively, which are much larger than the 0.064 dB, 0.048 dB and 0.050 dB increase by the DBF and SAO. This indicates the high fluctuation of frames after HEVC encoding at different configurations. Such high fluctuation of frame quality shows the potential to design an MIF that may significantly outperform the original in-loop filters in HEVC.

Besides quality fluctuation, content similarity is also a crucial factor in the proposed approach, considering that the motion compensation between a URF and its higher-quality reference frames typically works when they share similar content. The similarity is measured by calculating the correlation coefficient (CC) of luminance and chrominance matrices between two frames. Figure 3 shows the CC curves with standard deviation at various distance of frames in encoding order of HEVC, averaged on all the training and validation sequences. For both luminance (i.e., Y) and chrominance (i.e., U and V) channels, the CC is always positive, revealing the similarity in frame content. In addition, the three CC curves with standard deviation are similar, which implies a coherence of CC among Y, U and V channels. Although the frame distance is calculated based on the encoding order rather than displaying order, the increase of frame distance generally leads to decreasing CC. It can also be observed that in most cases the CC is larger than 0.7 within 10 frames, indicating a prominent similarity in frame content. Thus, adequate similar frames may always be obtained for a URF via searching from its adjacent frames.

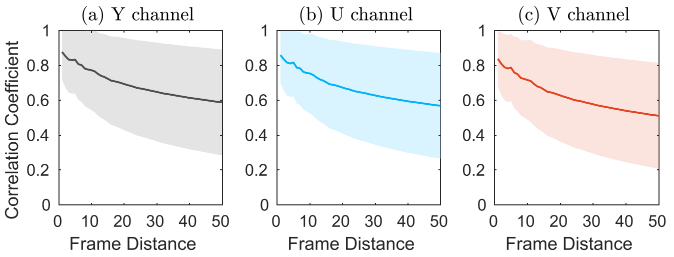


Fig. 3. Frame content similarity measured by CC at various frame distance. The line represents the average CC for each channel, and the colored foreground indicates the range within one standard deviation.

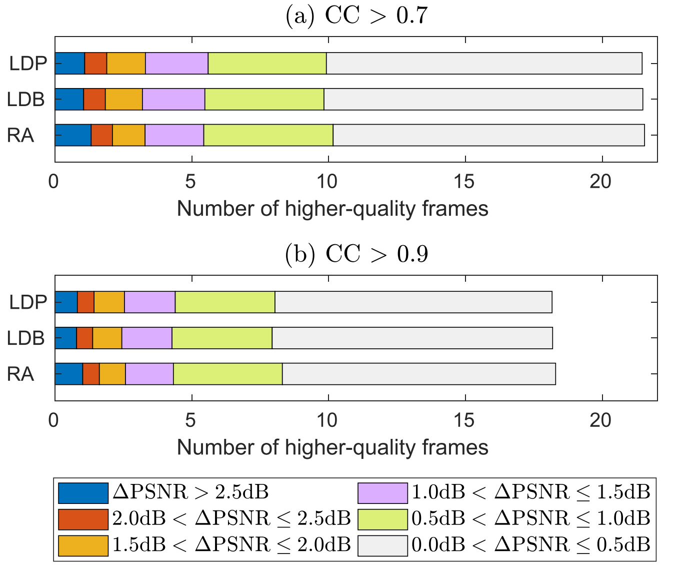


Fig. 4. Statistics of average number of higher-quality reference frames for each URF, satisfying CC larger than a certain threshold.

Finally, we analyze the composition of available reference frames for a URF in our MIF, considering both frame quality and content similarity. Figure 4-(a) counts the numbers of higher-quality frames for each URF in terms of PSNR increase, with both luminance and chrominance CC larger than a certain threshold. The statistics in this figure are averaged on all inter-predicted frames in the training and validation sequences at four QP values {22, 27, 32, 37}. It can be found that on average 9.9, 9.8 and 10.2 previously encoded frames with CC $>$ 0.7 and ΔPSNR $>$ 0.5dB are available for a URF, under the LDP, LDB and RA configurations, respectively. With a tighter constraint of CC $>$ 0.9, there are still 8.0, 7.9 and 8.3 previously encoded frames available with ΔPSNR $>$ 0.5dB under the three configurations, respectively. It can be expected that, the reference frames of substantially higher quality (e.g., ΔPSNR $>$ 2dB) are more helpful. Moreover, Figure 5 illustrates the subjective quality of reference frames for a URF, taking the 69th encoded frame of sequence *BalloonRising* at QP $=$ 32 as an example. As shown in this figure, totally 42 higher-quality reference frames with CC $>$ 0.7 are available for the URF, and typically the higher PSNR also corresponds

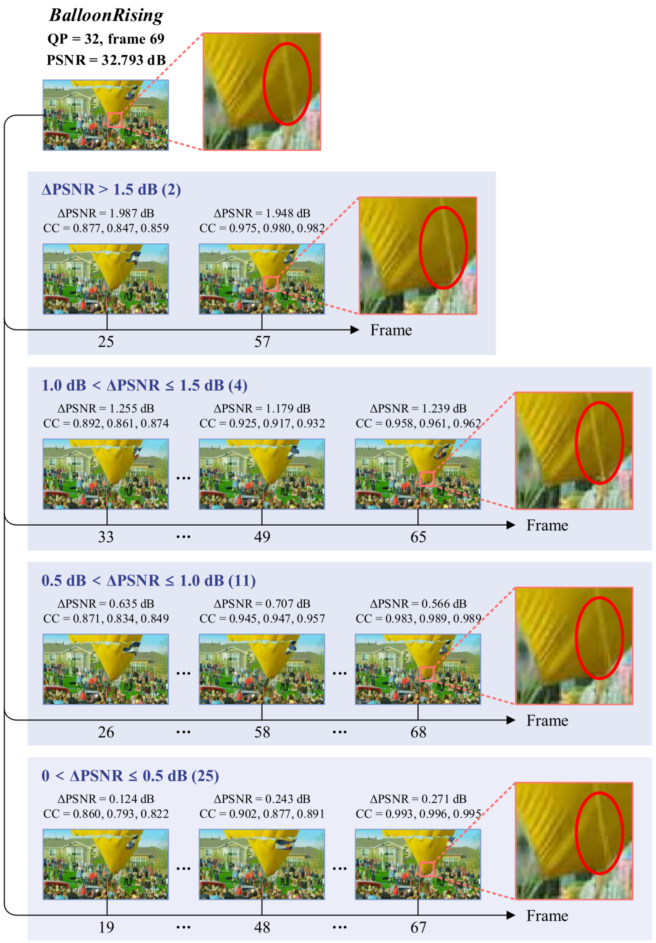


Fig. 5. Subjective examination of higher-quality reference frames with CC > 0.7 for a URF. The reference frames are grouped by different ranges of ΔPSNR, and the blue font in bracket represents the number of frames for each group. In each frame, the CC values are calculated for Y, U and V channels, respectively.

to the better subjective visual perception, especially in textured and moving regions (e.g., ropes attached to the hot air balloon). Also, in terms of content, the frames shown in Figure 5 appear to be similar. Therefore, based on both objective and subjective examination, it can be reasonably expected to find an adequate number of reference frames of much higher quality and similar to a URF.

IV. Proposed MIF Approach

A. Framework

The framework of our MIF approach is illustrated in Figure 6. In the standard HEVC, each raw frame is encoded through intra/inter-mode prediction, discrete transform and quantization. Then, the predicted frame and the residual frame form a URF. Subsequently, the URF is filtered with DBF and SAO for quality enhancement. Different from the standard HEVC, we propose a deep-learning-based in-loop filter to enhance the URF, leveraging information from its neighboring frames. First, RFS selects high-quality and high-correlated frames as reference, to be introduced in Section IV-B. Next, one of the two possible filtering modes is applied to the URF, as described below.

- **Mode 1: MIF-Net.** Assume that M reference frames are needed in MIF-Net. If RFS selects at least M frames, the URF is processed by MIF-Net to generate an enhanced frame. MIF-Net consists of two parts, i.e., motion compensation and quality enhancement. In MIF-Net, each reference frame is first aligned with the URF in terms of content, with a motion compensation network. Then, all the aligned reference frames and the URF are fed into a quality enhancement network to output the reconstructed frame, utilizing both spatial and temporal correlation of these frames. Note that the two networks are combined into an end-to-end model, which can be efficiently optimized with intermediate training.
- **Mode 2: IF-Net.** In the case that no enough reference frames are found for the URF, another deep neural network, IF-Net, is used instead as a simpler counterpart of MIF-Net. In contrast to MIF-Net, IF-Net only takes the URF as input without any consideration of multiple frames. The architecture of IF-Net is similar to that of the quality enhancement network in MIF-Net, and thus most training parameters in IF-Net can be initialized by the trained parameters in MIF-Net. Such design improves the effectiveness of training procedure because it is not necessary to train IF-Net from scratch.

In Modes 1 and 2, both MIF-Net and IF-Net are adaptive to the CU and TU partition, in which the parameters of the convolutional kernels are varied with respect to CU and TU partition. More details about the architectures of MIF-Net and IF-Net are introduced in Sections IV-C and IV-D, respectively, and the training protocol is presented in Section IV-E. If MIF-Net/IF-Net fails to improve frame quality, the standard DBF and SAO can also be used as a supplementary mode. Finally, the best mode among the three possible choices (i.e., MIF-Net, IF-Net and the standard in-loop filters) is selected as the actual choice, ensuring the overall performance of our approach.

B. Design of RFS

In our approach, RFS is designed to select the reference frames for each URF, serving as a basis of MIF. For the n-th URF (denoted by $\mathbf{F}_n^{\mathrm{U}}$) in a video sequence, RFS examines its previous N encoded frames as the reference frame pool, each of which is denoted by $\mathbf{F}_i^{\mathrm{P}}$ $(n - N \leq i \leq n - 1)$. Afterwards, six metrics reflecting quality difference and content similarity are calculated, as shown below.

- $\Delta\mathrm{PSNR}_{i,n}^{\mathrm{Y}}$, $\Delta\mathrm{PSNR}_{i,n}^{\mathrm{U}}$ and $\Delta\mathrm{PSNR}_{i,n}^{\mathrm{V}}$: PSNR increase of $\mathbf{F}_i^{\mathrm{P}}$ over $\mathbf{F}_n^{\mathrm{U}}$, for the Y, U and V channels, respectively.
- $\mathrm{CC}_{i,n}^{\mathrm{Y}}$, $\mathrm{CC}_{i,n}^{\mathrm{U}}$ and $\mathrm{CC}_{i,n}^{\mathrm{V}}$: the CC values of frame content between $\mathbf{F}_i^{\mathrm{P}}$ and $\mathbf{F}_n^{\mathrm{U}}$ for the Y, U and V channels.

Based on the above metrics, the procedure of RFS is shown in Figure 7. RFS first divides the reference frame pool into valid and invalid reference frames, and then all valid reference frames are fed into RFS-Net to select M frames used for enhancing the visual quality of $\mathbf{F}_n^{\mathrm{U}}$. To be more specific, a binary value $V_{i,n}$ represents whether a reference frame from the pool is valid. For at least one channel of $\mathbf{F}_i^{\mathrm{P}}$, if the PSNR increase is positive and the CC value is above a threshold τ,

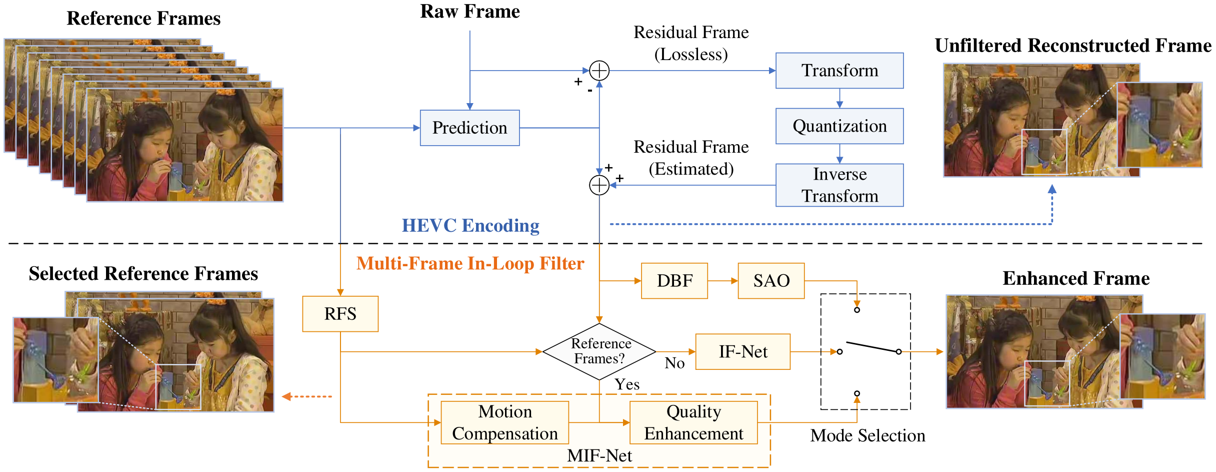


Fig. 6. Framework of the proposed MIF.

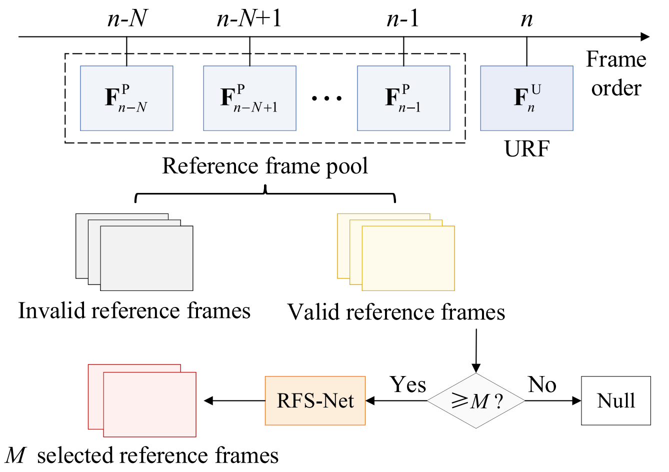

n-N
n-N+1
n-1
n
$\mathbf{F}^{\text{P}}_{n-N}$
$\mathbf{F}^{\text{P}}_{n-N+1}$
$\mathbf{F}^{\text{P}}_{n-1}$
$\mathbf{F}^{\text{U}}_{n}$
Frame order
URF
Reference frame pool
Invalid reference frames
Valid reference frames
Yes
No
RFS-Net
$\geqslant M$?
Null
M selected reference frames

Fig. 7. Procedure of RFS.

i.e., $V_{i,n} = 1$ in (1), $\mathbf{F}^{\text{P}}_i$ is treated as a valid reference frame.

$$V_{i,n} = \begin{cases} 1, & \text{if } \bigvee\limits_{c\in\{\text{Y,U,V}\}} (\Delta\text{PSNR}^c_{i,n} > 0 \wedge \text{CC}^c_{i,n} > \tau) \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

If there exist at least M valid reference frames, the six metrics for each valid reference frame $\mathbf{F}^{\text{P}}_i$ satisfying $n - N \leq i \leq n-1$ and $V_{i,n} = 1$ form a 6-dimensional vector, and then they are input to a two-layer fully connected network (termed by RFS-Net) to generate a scalar $\hat{R}_{i,n}$ as output, illustrated in Figure 8. Here, $\hat{R}_{i,n}$ is a continuous variable representing the potential of $\mathbf{F}^{\text{P}}_i$ being the reference for $\mathbf{F}^{\text{U}}_n$. A larger $\hat{R}_{i,n}$ indicates that $\mathbf{F}^{\text{P}}_i$ has more potential than other reference frames for enhancing $\mathbf{F}^{\text{U}}_n$. Note that $\hat{R}_{i,n}$ is the predicted value by RFS-Net, with the corresponding ground-truth value denoted by $R_{i,n}$.

The procedure to generate $R_{i,n}$ and train RFS-Net is presented in the following. Different from a randomly selected training batch for a typical neural network, the samples in one training batch for RFS-Net are extracted from the valid reference frames for only one URF. Such organization of samples is on account that all predicted values $\{\hat{R}_{i,n} | n - N \leq i \leq n-1, V_{i,n} = 1\}$ by RFS are used for enhancing one certain URF $\mathbf{F}^{\text{U}}_n$, without any consideration of other URFs.

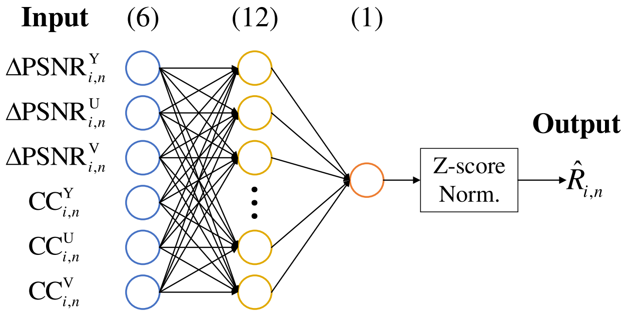

Input
(6)
(12)
(1)
$\Delta\text{PSNR}^{\text{Y}}_{i,n}$
$\Delta\text{PSNR}^{\text{U}}_{i,n}$
$\Delta\text{PSNR}^{\text{V}}_{i,n}$
$\text{CC}^{\text{Y}}_{i,n}$
$\text{CC}^{\text{U}}_{i,n}$
$\text{CC}^{\text{V}}_{i,n}$
Output
Z-score Norm.
$\hat{R}_{i,n}$

Fig. 8. Illustration of RFS-Net. The input is a 6-dimensional vector representing PSNR increase and CC of frame content, followed by 12 hidden nodes and 1 output node generated by the two layers in sequence. Both layers are activated with parametric rectified linear units (PReLU) [45], and the output of all samples in the same batch is processed with Z-score normalization, for obtaining the normalized $\hat{R}_{i,n}$.

In RFS-Net, the ground-truth potential $\{R_{i,n} | n - N \leq i \leq n-1, V_{i,n} = 1\}$ should reflect the quality of valid reference frames after these frames are aligned with $\mathbf{F}^{\text{U}}_n$. To achieve the content alignment, we apply the motion compensation network in MIF-Net to each valid reference frame $\mathbf{F}^{\text{P}}_i$ for $\mathbf{F}^{\text{U}}_n$ (satisfying $n - N \leq i \leq n-1$ and $V_{i,n} = 1$) to generate a compensated frame $\mathbf{F}^{\text{C}}_i$. Then, the difference between $\mathbf{F}^{\text{C}}_i$ and the n-th raw frame (denoted by $\mathbf{F}_n$) is able to quantify $R_{i,n}$, i.e., the ground-truth potential of $\mathbf{F}^{\text{P}}_i$ for enhancing $\mathbf{F}^{\text{U}}_n$. Here, normalized PSNR is used to calculate $R_{i,n}$ for each valid reference frame, formulated as

$$R_{i,n} = \frac{\text{PSNR}(\mathbf{F}^{\text{C}}_i, \mathbf{F}_n) - \mu^{\text{PSNR}}(\mathbf{F}_n)}{\sigma^{\text{PSNR}}(\mathbf{F}_n)}, \tag{2}$$

where PSNR$(\cdot,\cdot)$ denotes the PSNR between a compensated frame and its corresponding raw frame, and $\mu^{\text{PSNR}}(\mathbf{F}_n)$ and $\sigma^{\text{PSNR}}(\mathbf{F}_n)$ denote the mean value and the standard deviation over $\{\text{PSNR}(\mathbf{F}^{\text{C}}_i, \mathbf{F}_n) | n - N \leq i \leq n-1, V_{i,n} = 1\}$, respectively. After normalization, the ground-truth values in one batch are with the mean value of 0 and the standard deviation of 1, in accord with the normalized predicted values $\{\hat{R}_{i,n} | n - N \leq i \leq n-1, V_{i,n} = 1\}$. Therefore, $R_{i,n}$ and $\hat{R}_{i,n}$ are of similar scale, and the ℓ_2-loss can be used to measure the difference between them. Considering the whole training batch for

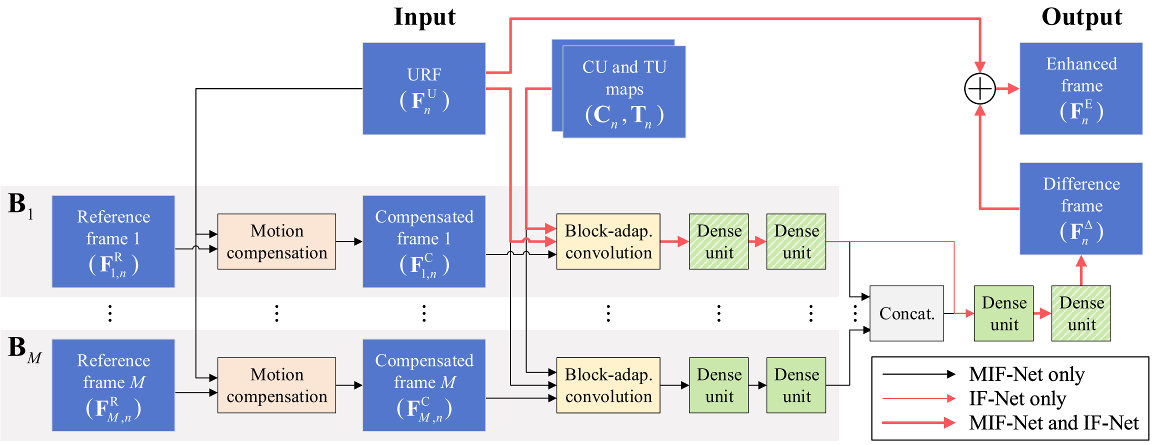


Fig. 9. Architecture of MIF-Net/IF-Net. The difference between MIF-Net and IF-Net is shown by different colors of arrows. Note that the background slashes in dense units indicate that the parameters of IF-Net can be initialized with those from MIF-Net.

enhancing $\mathbf{F}_n^{\mathrm{U}}$, the loss function of RFS-Net is formulated as

$$L_{\mathrm{RFS}} = \sum_{\substack{n-N \leq i \leq n-1 \\ V_{i,n}=1}} (R_{i,n} - \hat{R}_{i,n})^2, \tag{3}$$

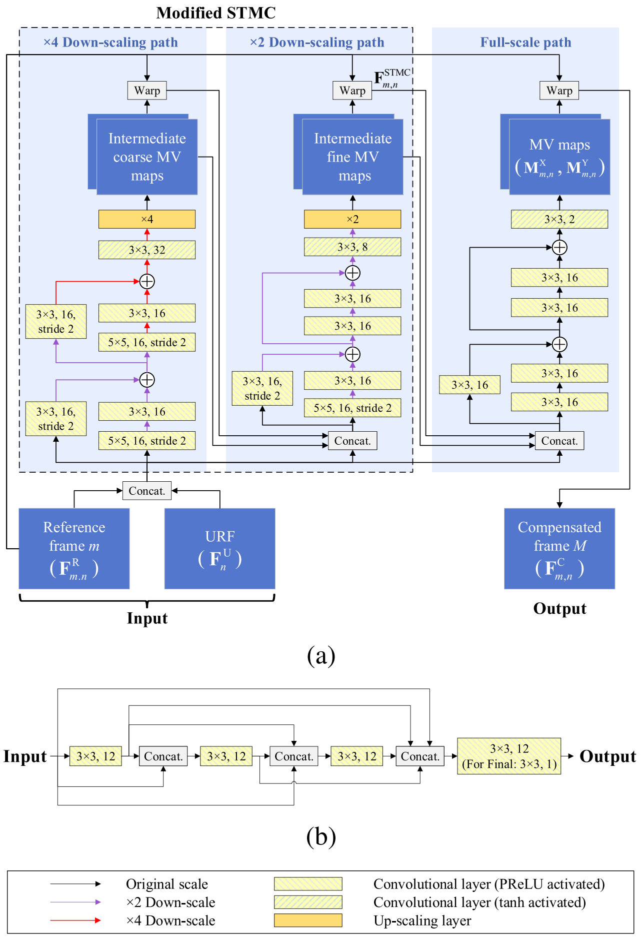
which is optimized by the Adam algorithm [44]. Using the trained RFS-Net model, the reference potential for all the valid frames $\{\hat{R}_{i,n} | n - N \leq i \leq n - 1, V_{i,n} = 1\}$ can be obtained. Then RFS selects M frames as output, denoted by $\{\mathbf{F}_{m,n}^{\mathrm{R}}\}_{m=1}^{M}$, where the index m indicates that $\mathbf{F}_{m,n}^{\mathrm{R}}$ is the frame with the m-th highest $\hat{R}_{i,n}$ among all valid reference frames. In the exceptional case that the number of valid reference frames is less than M, RFS does not work and IF-Net is used to enhance $\mathbf{F}_n^{\mathrm{U}}$ instead.

C. Architecture of MIF-Net

In our approach, the quality of each URF is enhanced by either MIF-Net or IF-Net, depending on the number of frames selected by RFS. This section mainly focuses on the architecture of MIF-Net, and the difference between IF-Net and MIF-Net is to be specified in Section IV-D. Figure 9 illustrates the overall architecture of MIF-Net/IF-Net. As shown in this figure, MIF-Net takes a URF $\mathbf{F}_n^{\mathrm{U}}$ and its M reference frames $\{\mathbf{F}_{m,n}^{\mathrm{R}}\}_{m=1}^{M}$ as the input, to generate the enhanced frame $\mathbf{F}_n^{\mathrm{E}}$ as the output. MIF-Net synthesizes information from M parallel branches $\{\mathbf{B}_m\}_{m=1}^{M}$, with each branch $\mathbf{B}_m$ dealing with the corresponding reference frame $\mathbf{F}_{m,n}^{\mathrm{R}}$. In branch $\mathbf{B}_m$, the reference frame $\mathbf{F}_{m,n}^{\mathrm{R}}$ is first aligned with the URF $\mathbf{F}_n^{\mathrm{U}}$ via a motion compensation network, to produce a compensated frame, denoted by $\mathbf{F}_{m,n}^{\mathrm{C}}$. Next, $\mathbf{F}_n^{\mathrm{U}}$ and $\mathbf{F}_{m,n}^{\mathrm{C}}$ are processed with a novel convolutional layer guided by the CTU partitioning structure of $\mathbf{F}_n^{\mathrm{U}}$ (named block-adaptive convolutional layer), to explore low-level features from different sources and merge the features with consideration of the CU and TU partition. Then, the low-level features flow through two successive DenseNet-based units (named dense units) [28] to extract more comprehensive features within $\mathbf{B}_m$. Finally, the extracted features from all the M branches are concatenated together and further processed with two dense units to extract high-level features. For ease of training, the output of the last dense unit (denoted by $\mathbf{F}_n^{\Delta}$) is regarded as a difference frame, and the enhanced frame $\mathbf{F}_n^{\mathrm{E}}$ is the summation of $\mathbf{F}_n^{\Delta}$ and $\mathbf{F}_n^{\mathrm{U}}$. The details of MIF-Net components are presented in the following.

Motion compensation network. In general, the content of a reference frame $\mathbf{F}_{m,n}^{\mathrm{R}}$ differs from that of $\mathbf{F}_n^{\mathrm{U}}$ due to temporal motion across frames. Therefore, we propose a motion compensation network based on the spatial transformer motion compensation (STMC) model [26], for content alignment between $\mathbf{F}_{m,n}^{\mathrm{R}}$ and $\mathbf{F}_n^{\mathrm{U}}$, illustrated in Figure 10-(a). In [26], the STMC model takes both $\mathbf{F}_{m,n}^{\mathrm{R}}$ and $\mathbf{F}_n^{\mathrm{U}}$ as input to obtain a compensated frame as output, denoted by $\mathbf{F}_{m,n}^{\mathrm{STMC}}$. The STMC consists of two paths (×4 and ×2 down-scaling paths) for predicting coarse and fine motion vector (MV) maps between the two input frames, respectively. Each path contains a succession of convolutional layers and an upscaling layer, and the fine MV maps from the ×2 down-scaling path are applied to $\mathbf{F}_{m,n}^{\mathrm{R}}$ for outputting $\mathbf{F}_{m,n}^{\mathrm{STMC}}$. In the STMC, the ×2 and ×4 down-sampling is capable for estimating various scales of motion. However, the accuracy of the STMC is limited due to down-sampling, and the architecture of the STMC can also be improved. To address this issue, we propose a motion compensation network, with the following advancements over the STMC in [26].

- Besides the ×2 and ×4 down-scaling paths, a full-scale path without down-sampling is added to enhance the precision of MV estimation. As shown in Figure 10-(a), $\mathbf{F}_{m,n}^{\mathrm{R}}$, $\mathbf{F}_n^{\mathrm{U}}$, $\mathbf{F}_{m,n}^{\mathrm{STMC}}$ and the ×2 MV maps from the STMC are concatenated together and input to this path. Afterwards, they are processed through convolutional layers to generate the final MV maps. All convolutional layers on this path are with stride of 1, keeping the size of feature maps the same as $\mathbf{F}_{m,n}^{\mathrm{R}}$ and $\mathbf{F}_n^{\mathrm{U}}$.
- Inspired by the ResNet [46], in total 6 shortcuts are added next to the convolutional layers for higher network capacity and ease to be trained. Note that both identity shortcuts and projection shortcuts are used, depending on the numbers/sizes of feature maps before and after each shortcut.
- All rectified linear units (ReLU) [47] activating convolutional layers in the STMC are replaced by PReLU, to adaptively learn the rectifying parameters [45].


Modified STMC
×4 Down-scaling path
×2 Down-scaling path
Full-scale path
Warp
Warp
Warp
$\mathbf{F}^{\mathrm{STMC}}_{m,n}$
Intermediate coarse MV maps
Intermediate fine MV maps
MV maps ($\mathbf{M}^{\mathrm{X}}_{m,n}$, $\mathbf{M}^{\mathrm{Y}}_{m,n}$)
×4
×2
3×3, 2
3×3, 32
3×3, 8
3×3, 16
3×3, 16, stride 2
5×5, 16, stride 2
Concat.
Reference frame m ($\mathbf{F}^{\mathrm{R}}_{m,n}$)
URF ($\mathbf{F}^{\mathrm{U}}_{n}$)
Compensated frame M ($\mathbf{F}^{\mathrm{C}}_{m,n}$)
Input
Output
3×3, 12
3×3, 12 (For Final: 3×3, 1)
Original scale
×2 Down-scale
×4 Down-scale
Convolutional layer (PReLU activated)
Convolutional layer (tanh activated)
Up-scaling layer

(a)

(b)

Fig. 10. Network details. (a) Motion compensation network. (b) Dense unit. For convolutional layers, "$p \times p$, q" represents q output channels with $p \times p$ kernels. Note that the convolutional stride is set to 1 by default, except that explicitly mentioned in certain layers.

With the above modifications, the full-scale path outputs two MV maps, $\mathbf{M}^{\mathrm{X}}_{m,n}$ and $\mathbf{M}^{\mathrm{Y}}_{m,n}$, denoting the horizontal and vertical motion of all pixels from $\mathbf{F}^{\mathrm{R}}_{m,n}$ to $\mathbf{F}^{\mathrm{U}}_{n}$. Finally, the compensated frame $\mathbf{F}^{\mathrm{C}}_{m,n}$ is derived by

$$\mathbf{F}^{\mathrm{C}}_{m,n}(x,y)=\mathrm{Bil}\{\mathbf{F}^{\mathrm{R}}_{m,n}(x+\mathbf{M}^{\mathrm{X}}_{m,n}(x,y), y+\mathbf{M}^{\mathrm{Y}}_{m,n}(x,y))\}, \tag{4}$$

where x and y are coordinates of a pixel, and $\mathrm{Bil}\{\cdot\}$ represents the bilinear interpolation considering that the motion may be of non-integer pixels.

Block-adaptive convolutional layers. In each branch of MIF-Net, the compensated frame $\mathbf{F}^{\mathrm{C}}_{m,n}$ and the URF $\mathbf{F}^{\mathrm{U}}_{n}$ are processed with a convolutional layer adaptive to the CU and TU partition in HEVC. The input to this layer is a concatenation of three feature maps, including $\mathbf{F}^{\mathrm{C}}_{m,n}$, $\mathbf{F}^{\mathrm{U}}_{n}$ and $\mathbf{F}^{\mathrm{C}}_{m,n}-\mathbf{F}^{\mathrm{U}}_{n}$. In addition to $\mathbf{F}^{\mathrm{C}}_{m,n}$ and $\mathbf{F}^{\mathrm{U}}_{n}$, $\mathbf{F}^{\mathrm{C}}_{m,n}-\mathbf{F}^{\mathrm{U}}_{n}$ is also meaningful. It is because $\mathbf{F}^{\mathrm{C}}_{m,n}-\mathbf{F}^{\mathrm{U}}_{n}$ reflects the reliability of $\mathbf{F}^{\mathrm{C}}_{m,n}$ as a reference frame for $\mathbf{F}^{\mathrm{U}}_{n}$, since an overlarge distance between two co-located parts of $\mathbf{F}^{\mathrm{C}}_{m,n}$ and $\mathbf{F}^{\mathrm{U}}_{n}$ may indicate ineffective motion compensation at these parts. In this layer, the CU and TU partition for $\mathbf{F}^{\mathrm{U}}_{n}$ is represented by two feature maps, i.e., $\mathbf{C}_n$ and $\mathbf{T}_n$, respectively. The size of $\mathbf{C}_n$ or $\mathbf{T}_n$ is equal to that of $\mathbf{F}^{\mathrm{U}}_{n}$, and the values in each map are assigned according to the partition structure. If pixel (x, y) is on the boundary of a CU or TU, the corresponding value $\mathbf{C}_n(x, y)$ or $\mathbf{T}_n(x, y)$ is set to 1. Otherwise, the value is set to -1. Afterwards, the target of a block-adaptive convolutional layer is to output a certain number of feature maps, providing three feature maps as the input and two feature maps as the guidance. For this problem, we present a guided convolution operation in Algorithm 1, assuming that P^{I}, P^{G} and P^{O} feature maps are used as the input, guidance and output, respectively. This algorithm consists of two main procedures:

- Intermediate map extraction (line 1): to extract various features from the guidance features maps, meanwhile ensuring the number of intermediate feature maps equal to the output channels P^{O}.
- Convolution with intermediation (lines 2∼9): to conduct the convolution in which the weights are adaptively adjusted according to the intermediate feature maps.

Compared with a typical convolutional layer where the space-irrelevant weights are shared across the whole feature map, the major advancement of this algorithm lies in the space-relevant weights generated according to the guidance (see line 5 in Algorithm 1), contributing to a higher network capacity. Moreover, because of only convolution rather than full-connection is added for intermediate map extraction, the number of trainable parameters is not sharply increased, which results in little risk of over-fitting. For each block-adaptive convolutional layer in MIF-Net, $P^{\mathrm{I}}=3$ and $P^{\mathrm{G}}=2$ as described above, and the number of output maps is set to be $P^{\mathrm{O}}=12$.

Dense units for quality enhancement. In [28], Huang *et. al.* have proposed an efficient variant of CNN, named DenseNet, which introduces different length of connections between the input and output. Compared with a plain CNN or the ResNet [46], the efficiency of DenseNet to train a deep network mainly results from the alleviation of vanishing gradients, the encouragement of feature reuse and the reduction of computational complexity. Considering these compelling advantages, totally $(2M+2)$ dense units are adopted in our MIF-Net for quality enhancement, i.e., 2 dense units in each branch and 2 dense units at the end of MIF-Net synthesizing features from all the M branches. Here, all the dense units are with the same structure, as illustrated in Figure 10-(b). Each dense unit contains 4 convolutional layers, and before each layer, features from all preceding layers are concatenated together. Thus, a dense unit includes 10 inter-layer connections, much more than a 4-layer plain CNN with only 4 connections. At each layer in the dense unit, the number of output channels is 12, except the last layer in the final dense unit which outputs only 1 channel as the difference frame between $\mathbf{F}^{\mathrm{E}}_{n}$ and $\mathbf{F}^{\mathrm{U}}_{n}$.

D. Difference Between MIF-Net and IF-Net

The previous section has elaborated our MIF-Net in detail. In the following, we introduce the IF-Net as a simpler counterpart of MIF-Net, adopted in the case that no enough reference frames are found for the URF. The only difference between IF-Net and MIF-Net lies in the absence of M reference frames

Algorithm 1 Guided Convolution

```
Input:
    $\{\mathbf{F}_j^{\mathrm{I}}\}_{j=1}^{P^{\mathrm{I}}}$: Input feature maps. $\{\mathbf{F}_k^{\mathrm{G}}\}_{k=1}^{P^{\mathrm{G}}}$: Feature maps as the guidance.
    $\{w_{j,l}(\Delta x, \Delta y) | 1 \leq j \leq P^{\mathrm{I}}, 1 \leq l \leq P^{\mathrm{O}}, -1 \leq \Delta x, \Delta y \leq 1\}$: Convolutional weights. Each element is the weight in the 3×3 kernel from the $j$-th input map to the $l$-th output map, where $\Delta x$ and $\Delta y$ are the relative indices within the kernel.
Output: $\{\mathbf{F}_l^{\mathrm{O}}\}_{l=1}^{P^{\mathrm{O}}}$: Output feature maps.
Process $\{\mathbf{F}_k^{\mathrm{G}}\}_{k=1}^{P^{\mathrm{G}}}$ with two typical convolutional layers to generate $P^{\mathrm{O}}$ intermediate feature maps, denoted by $\{\mathbf{F}_l^{\mathrm{M}}\}_{l=1}^{P^{\mathrm{O}}}$. Each layer outputs $P^{\mathrm{O}}$ channels, convoluted by 3×3 kernels with stride of 1 and activated by PReLU.
for $1 \leq l \leq P^{\mathrm{O}}$ do
    for $1 \leq x \leq$ width of $\mathbf{F}_l^{\mathrm{O}}$ and $1 \leq y \leq$ height of $\mathbf{F}_l^{\mathrm{O}}$ do
        for $\Delta x \in \{-1, 0, 1\}$ and $\Delta y \in \{-1, 0, 1\}$ do
            Compute the modified weight guided by $\mathbf{F}_l^{\mathrm{M}}$: $w_{j,l}^{\mathrm{G}}(\Delta x, \Delta y) = w_{j,l}(\Delta x, \Delta y) \cdot \mathbf{F}_l^{\mathrm{M}}(x + \Delta x, y + \Delta y)$. *
        end for
        Compute the output pixel: $\mathbf{F}_l^{\mathrm{O}}(x, y) = \sum_{j=1}^{P^{\mathrm{I}}} \sum_{\Delta x=-1}^{1} \sum_{\Delta y=-1}^{1} w_{j,l}^{\mathrm{G}}(\Delta x, \Delta y) \cdot \mathbf{F}_j^{\mathrm{I}}(x + \Delta x, y + \Delta y)$ *
    end for
end for
* If $(x + \Delta x, y + \Delta y)$ is beyond the frame size, the zero padding is adopted.
```

for $\mathbf{F}_n^{\mathrm{U}}$ in IF-Net. Therefore, only the quality enhancement network without motion compensation is adopted in IF-Net, as illustrated with the red arrows in Figure 9. Compared with MIF-Net, only one branch $\mathbf{B}_1$ without the compensated frame exists in IF-Net, and the concatenation synthesizing M branches is omitted. Despite the simpleness, a guided convolutional layer and four consecutive dense units still exist in IF-Net, ensuring sufficient network capacity for quality enhancement.

E. Training of MIF-Net and IF-Net

With both motion compensation and quality enhancement in M branches, MIF-Net is an end-to-end deep neural network that may be difficult to be trained by directly minimizing the difference between $\mathbf{F}_n^{\mathrm{E}}$ and $\mathbf{F}_n$. To solve this problem, we propose to train MIF-Net with intermediate supervision [48], via introducing two loss functions into MIF-Net to optimize the whole network at different stages. First, the difference between $\mathbf{F}_n^{\mathrm{U}}$ and each of M compensated frames $\mathbf{F}_{m,n}^{\mathrm{C}}$ can measure the performance of the motion compensation network, and thus it is defined as the intermediate loss

$$L_{\mathrm{INT}} = \frac{1}{M} \sum_{m=1}^{M} \|\mathbf{F}_{m,n}^{\mathrm{C}} - \mathbf{F}_n^{\mathrm{U}}\|_2^2, \tag{5}$$

where $\|\cdot\|_2$ represents the ℓ_2-norm difference between two frames. Next, the ℓ_2-norm difference between $\mathbf{F}_n^{\mathrm{E}}$ and $\mathbf{F}_n$ indicates the performance of the whole MIF-Net, and the global loss is defined as

$$L_{\mathrm{GLO}} = \|\mathbf{F}_n^{\mathrm{E}} - \mathbf{F}_n\|_2^2. \tag{6}$$

Combining the above two loss functions, the loss L for our MIF-Net is the weighted summation of them, formulated as

$$L = \alpha \cdot L_{\mathrm{INT}} + \beta \cdot L_{\mathrm{GLO}}. \tag{7}$$

Here, α and β are changeable weights, and L is optimized by the Adam algorithm [44]. On account that the optimal performance of quality enhancement relies on the well-trained motion compensation network, the intermediate loss L_{INT} should be optimized with a larger weight by setting $\alpha \gg \beta$ at early stage of training. After L_{INT} converges, we set $\beta \gg \alpha$ instead, in order to emphasize more on optimization of the global loss L_{GLO}. Through the two stages of training, the URF $\mathbf{F}_n^{\mathrm{U}}$ can be significantly enhanced using M selected reference frames. In contrast to MIF-Net, the training procedure of IF-Net is easier, considering the absence of motion compensation. In IF-Net, the trainable parameters in three dense units can be initialized by those in well-trained MIF-Net, with no need to train from scratch. In addition, the loss of IF-Net is the same as L_{GLO} in MIF-Net, which can be directly optimized by the Adam.

V. Syntax Regulation

In our MIF approach, some control data about RFS and filtering mode selection should be shared by both the encoder and decoder. Therefore, the corresponding syntax is regulated for each URF $\mathbf{F}_n^{\mathrm{U}}$, as listed in Table III. The details of syntax regulation are presented in the following.

Syntax for RFS. First, *MIF_Net_on* signals whether MIF-Net is adopted, depending on the number of reference frames selected by RFS. If *MIF_Net_on* is *true*, *Ref_index* is activated to represent the indices of M reference frames, selected from the N-frame pool by RFS. To save the bit-rate, only the distance between a reference frame and $\mathbf{F}_n^{\mathrm{U}}$ (i.e., $|i - n|$ satisfying $n - N \leq i \leq n - 1$) is encoded, rather than encoding the absolute frame index n. Consequently, the number of bits for encoding *Ref_index* is $M \lceil \log_2 N \rceil$.

Syntax for filtering mode selection. In addition to RFS, the selection of filtering mode is also encoded to signal whether the URF is processed by a proposed network or by the standard in-loop filters. For each channel $c \in \{\mathrm{Y,U,V}\}$, assume that the size of a frame is $W_c \times H_c$. Considering that the same filtering mode may not be suitable for different patches of a frame, the selection is conducted in $p_c \times p_c$ patches. Here, a frame can be divided into $\left\lceil \frac{W_c}{p_c} \right\rceil \cdot \left\lceil \frac{H_c}{p_c} \right\rceil$ non-overlapping patches, and the syntax *Mode_c*$[u][v]$ denotes whether a proposed network is used for the (u, v)-th patch. If the quality of a patch processed by MIF-Net or IF-Net is higher than that filtered by the standard DBF and SAO, *Mode_c*$[u][v]$

TABLE III

SYNTAX REGULATION

Syntax name	Number of bits
MIF_Net_on	1
$\{Ref\_index[m]\}_{m=1}^{M}$ (when *MIF_Net_on* is *true*)	$M\lceil\log_2 N\rceil$
$\{\{Mode\_Y[u][v]\}_{u=1}^{\lceil\frac{H_{\mathrm{Y}}}{p_{\mathrm{Y}}}\rceil}\}_{v=1}^{\lceil\frac{W_{\mathrm{Y}}}{p_{\mathrm{Y}}}\rceil}$ $\{\{Mode\_U[u][v]\}_{u=1}^{\lceil\frac{H_{\mathrm{U}}}{p_{\mathrm{U}}}\rceil}\}_{v=1}^{\lceil\frac{W_{\mathrm{U}}}{p_{\mathrm{U}}}\rceil}$ $\{\{Mode\_V[u][v]\}_{u=1}^{\lceil\frac{H_{\mathrm{V}}}{p_{\mathrm{V}}}\rceil}\}_{v=1}^{\lceil\frac{W_{\mathrm{V}}}{p_{\mathrm{V}}}\rceil}$	$\sum_{c\in\{\mathrm{Y,U,V}\}}\left\lceil\frac{W_c}{p_c}\right\rceil\cdot\left\lceil\frac{H_c}{p_c}\right\rceil$

Note: $\lceil\cdot\rceil$ represents the top integral function.

TABLE IV

HYPER-PARAMETERS FOR NETWORKS

Hyper-parameter	RFS-Net	MIF-Net or IF-Net
Size of ref. frame pool in RFS: N	16	-
Threshold for CC value in RFS: τ	0.3	-
Num. of selected reference frames: M	2	
Optimization	Xavier initialization* [51] and Adam [44]	
Batch size	$\leq 16^{**}$	16
Initial learning rate	10^{-5}	10^{-4}
Num. of iterations	10^5	10^6 (from scratch) 2×10^5 (fine-tunning)
Changeable weights in MIF-Net: α and β	-	0.99 & 0.01 (at beginning) 0.01 & 0.99 (after L_{INT} converged)

* Except that the parameters in three dense units of IF-Net are initialized with those from MIF-Net, as mentioned in Section IV-E.

** The batch size equals to the number of valid reference frames for a URF.

is set to *true*, and otherwise set to *false*. Considering the adjustable patch width, a smaller p_c indicates more refined mode selection prone to better frame quality, but introduces more bit-rate redundancy due to the encoded bits of the syntax *Mode_c*. In contrast, a larger p_c means fewer bits for the syntax, while leading to lower frame quality. Therefore, there exists a trade-off to choose a reasonable p_c, and the value of p_c is discussed in Section VI-A.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our MIF approach through experimental results. Section VI-A presents the settings in the experiments. In Section VI-B, we evaluate both objective and subjective performance of our MIF approach at the RA configuration, compared with the HM baseline and two state-of-the-art approaches, [10] and [20]. In Section VI-C, we further verify the effectiveness and generalization ability of our MIF approach with various settings. Finally, the ablation study is conducted in Section VI-D.

A. Settings

Experimental configurations. In our experiments, all approaches for in-loop filtering were incorporated into the HEVC reference software HM 16.5 [43]. The RA configuration was applied using the file *encoder_randomaccess_main.cfg* [49] for both network training and performance evaluation at four QPs, $\{22, 27, 32, 37\}$. The 120 training sequences in our HIF database were used to train the networks, and the hyper-parameters were tuned over the 40 validation sequences. Note that all video sequences are in YUV format. During the training phase, only the Y channel was input to both MIF-Net and IF-Net. It is because the Y channel is luminance that contains most visual information. Therefore, during the test phase, the trained models on the Y channel were directly used on all three channels. In the test stage, we set the patch width (introduced in Section V) to be $p_{\mathrm{Y}} = p_{\mathrm{U}} = p_{\mathrm{V}} = 256$. For performance evaluation, the Bjøntegaard delta bit-rate (BD-BR) and Bjøntegaard delta PSNR (BD-PSNR) [50] were measured to assess the rate-distortion (RD) performance. The evaluation was conducted on 40 video sequences in total, containing all 18 sequences of the JCT-VC standard test set [42] and the 22 test sequences in our HIF database, named as the supplementary test set. Note that the test sequences were non-overlapping with both training and validation sequences. All experiments were conducted on a computer with an Intel (R) Core (TM) i7-7700K CPU @4.2 GHz, 32 GB RAM and the Ubuntu 16.04 (64-bit) operating system. In addition, a GeForce GTX 1080 GPU was used to accelerate the training procedure.

Network settings. For our approach, one MIF-Net model and one IF-Net model were trained for each evaluated QP, while all QPs shared the same trained RFS-Net model. The tuned hyper-parameters for these networks are listed in Table IV. For training the models of MIF-Net and IF-Net, all frames were segmented into 64×64 patches. Considering the efficiency of training, the IF-Net or MIF-Net model at QP $= 32$ was trained from scratch, while the models at QPs $\{22, 27, 37\}$ were fine-tuned from the trained model.

B. Performance Analysis

Objective RD performance. First, we evaluate the objective RD performance of our MIF approach in terms of the BD-BR and BD-PSNR, compared with the HM baseline (standard DBF and SAO), a heuristic approach [10] and a learning-based approach [20]. For a fair comparison, the models of [20] were re-trained on our HIF database. Table V tabulates the RD results of all four approaches, in which the original HM without the in-loop filter is used as an anchor. As indicated in Table V-(a), the BD-BR of our MIF approach is -11.621% averaged over the 18 standard test sequences, outperforming -5.031% of the DBF and SAO, -6.295% of [10] and -9.227% of [20]. In addition, Table V-(b) shows that the average BD-BR of our approach is -12.607% over the supplementary test set, and it also significantly outperforms those of other three approaches, i.e, -4.449% of the DBF and SAO, -5.746% of [10] and -9.942% of [20]. In terms of BD-PSNR, our approach achieves 0.391 dB in the standard test set and 0.502 dB in the supplementary test set, also considerably better than the DBF and SAO (0.162 dB and 0.167 dB), [10] (0.201 dB and 0.219 dB) and [20] (0.305 dB and 0.392 dB). In a word, our MIF approach achieves the best RD performance among all four approaches. The possible reasons of such outperformance include: (1) the utilization of multiple adjacent frames, (2) the effective dense blocks

TABLE V

RD PERFORMANCE OF IN-LOOP FILTERS (RA CONFIG.). (a) PERFORMANCE ON THE JCT-VC TEST SET. (b) PERFORMANCE ON THE SUPPLEMENTARY TEST SET

Class	Sequence	DBF and SAO		[10]		[20]		Proposed MIF	
		BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)
A	*PeopleOnStreet*	-8.287	0.368	-12.025	0.540	-12.484	0.568	**-16.824**	**0.777**
	Traffic	-5.348	0.162	-6.169	0.188	-9.816	0.304	**-12.152**	**0.383**
B	*BasketballDrive*	-6.655	0.148	-8.838	0.198	-11.053	0.250	**-14.870**	**0.347**
	BQTerrace	-7.149	0.111	-11.397	0.173	-14.365	0.228	**-17.126**	**0.271**
	Cactus	-7.540	0.157	-8.904	0.189	-12.518	0.272	**-15.829**	**0.349**
	Kimono	-7.536	0.220	-9.261	0.273	-10.482	0.311	**-12.235**	**0.368**
	ParkScene	-3.679	0.112	-4.083	0.124	-5.940	0.182	**-7.994**	**0.249**
C	*BasketballDrill*	-5.017	0.206	-5.393	0.222	-7.818	0.326	**-10.324**	**0.434**
	BQMall	-3.933	0.150	-4.451	0.170	-7.663	0.296	**-9.376**	**0.367**
	PartyScene	-1.054	0.044	-1.224	0.051	-2.417	0.100	**-4.159**	**0.173**
	RaceHorses	-6.154	0.222	-7.082	0.257	-10.403	0.386	**-12.736**	**0.476**
D	*BasketballPass*	-3.852	0.182	-4.324	0.204	-7.700	0.370	**-9.984**	**0.484**
	BlowingBubbles	-0.829	0.034	-0.831	0.034	-3.072	0.126	**-3.980**	**0.164**
	BQSquare	-0.053	0.002	0.010	-0.000	-3.263	0.123	**-4.401**	**0.165**
	RaceHorses	-4.441	0.199	-4.797	0.215	-8.860	0.407	**-10.992**	**0.510**
E	*FourPeople*	-7.018	0.262	-8.489	0.319	-13.937	0.538	**-16.480**	**0.644**
	Johnny	-5.599	0.143	-8.034	0.208	-11.649	0.302	**-14.370**	**0.378**
	KristenAndSara	-6.410	0.203	-8.011	0.254	-12.637	0.406	**-15.340**	**0.501**
Average		-5.031	0.162	-6.295	0.201	-9.227	0.305	**-11.621**	**0.391**

(a)

Resolution	Sequence	DBF and SAO		[10]		[20]		Proposed MIF	
		BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)
352×288	*Bowing*	-5.490	0.315	-6.514	0.375	-12.668	0.753	**-13.765**	**0.824**
	Bus	-1.789	0.085	-1.900	0.090	-5.696	0.276	**-7.132**	**0.346**
	Football	-4.340	0.226	-5.268	0.275	-8.933	0.479	**-11.604**	**0.631**
	Monitor	-7.280	0.195	-9.165	0.252	-14.959	0.420	**-17.888**	**0.503**
	News	-3.231	0.174	-3.596	0.194	-9.361	0.519	**-10.511**	**0.585**
640×360	*Fountain*	-2.767	0.134	-3.448	0.167	-5.202	0.255	**-6.940**	**0.343**
	Bridge	-3.358	0.197	-4.985	0.293	-8.588	0.515	**-10.741**	**0.649**
	Cars	-4.899	0.238	-5.874	0.287	-8.746	0.431	**-10.268**	**0.512**
	Dance	-5.680	0.219	-7.220	0.281	-10.413	0.410	**-13.828**	**0.554**
	Writing	-8.171	0.369	-12.105	0.560	-13.732	0.644	**-17.312**	**0.817**
720×480	*Garden*	-1.502	0.063	-1.950	0.082	-6.743	0.291	**-13.594**	**0.600**
	Galleon	-3.423	0.124	-3.702	0.135	-8.332	0.309	**-10.122**	**0.379**
	Calendar	-0.594	0.021	-0.948	0.034	-7.278	0.268	**-15.048**	**0.565**
	WashDC	-4.386	0.188	-6.966	0.305	-13.578	0.607	**-16.620**	**0.742**
1280×720	*Mobcal*	-2.732	0.066	-3.830	0.096	-6.689	0.164	**-8.053**	**0.206**
	Shields	-4.638	0.104	-5.165	0.120	-11.009	0.254	**-12.497**	**0.300**
	Holm	-7.545	0.129	-8.825	0.151	-15.224	0.266	**-18.519**	**0.335**
1920×1080	*OneDark*	-8.400	0.195	-11.221	0.267	-16.235	0.387	**-20.014**	**0.478**
	HongKong	-2.124	0.083	-2.421	0.095	-5.271	0.209	**-6.726**	**0.270**
	Stadium	-1.866	0.097	-2.581	0.135	-6.373	0.340	**-7.765**	**0.416**
	Raptors	-5.663	0.214	-7.356	0.280	-10.431	0.399	**-11.200**	**0.435**
	Diver	-7.989	0.245	-11.373	0.355	-13.257	0.417	**-17.210**	**0.556**
Average		-4.449	0.167	-5.746	0.219	-9.942	0.392	**-12.607**	**0.502**

(b)

and (3) the proposed block-adaptive convolutional layers. Their contributions in the RD gain are to be analyzed in Section VI-D.

Subjective visual quality. Next, we compare the subjective quality of all four approaches. Figure 11 illustrates some regions of compressed video sequences as examples, compressed at QP = 37 and the RA configuration. For *RaceHorses*, it can be observed that the edges of the horse tail are severely blurred when compressed by the DBF and SAO, [10] and [20]. In contrast, the horse tail is with clearer edges after being enhanced by our MIF approach. Also, on the pedestrians in *PeopleOnStreet* and the hand in *FourPeople*, the blocking artifacts are significantly reduced by our approach, compared with other three approaches. These examples show that our approach is probably with better visual quality of compressed videos, and the quality enhancement may be more observable at moving regions of frames. Moreover, we have also uploaded the bitstream files of 22 test sequences online,[3] encoded by both our MIF approach and the standard HEVC. With the corresponding decoders, the visual quality of all the frames can be observed for our MIF approach.

Time complexity. In addition, we analyze the complexity overhead introduced by our MIF approach and other adapted in-loop filters [10], [20]. First, the running time to encode one frame in HM [43], denoted by T_{HM}, is provided in Table VI. Based on this, we have also measured the time overhead T_{f} introduced by each in-loop filter and tabulated the ratio $\frac{T_{\mathrm{f}}}{T_{\mathrm{HM}}}$ in the rest of Table VI. A larger ratio $\frac{T_{\mathrm{f}}}{T_{\mathrm{HM}}}$ indicates relatively

[3] Available at: https://github.com/tianyili2017/HIF-Database

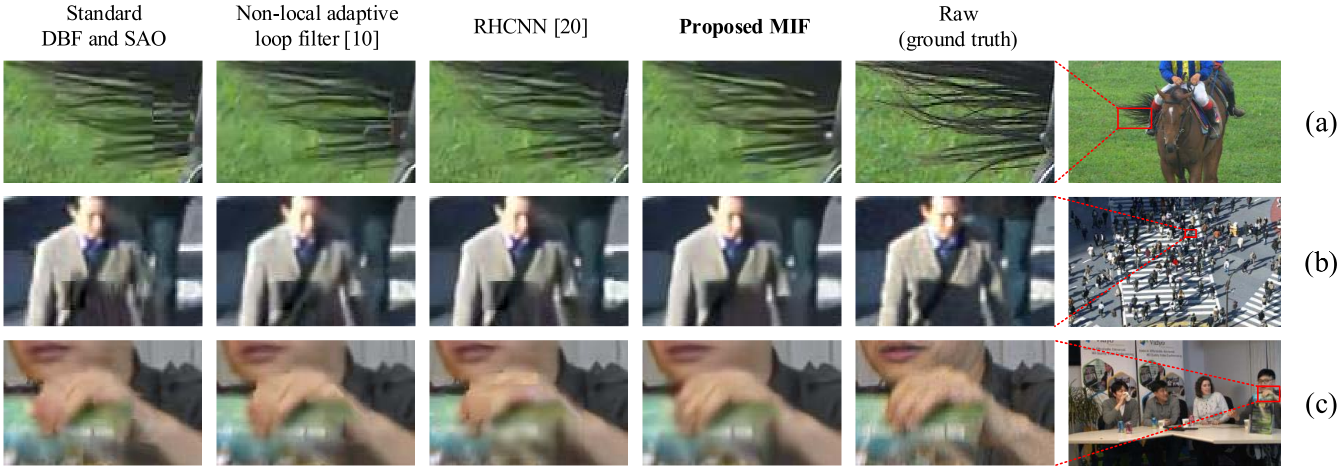


Fig. 11. Comparison of subjective visual quality. (a) *RaceHorses* (Class C). (b) *PeopleOnStreet* (Class A). (c) *FourPeople* (Class E).

TABLE VI
TIME COMPLEXITY OF IN-LOOP FILTERS (RA CONFIG.)

Resolution	T_{HM} (s)	$\frac{T_{\mathrm{f}}}{T_{\mathrm{HM}}}$				
		[10]	[20]		Proposed MIF	
			CPU	GPU	CPU	GPU
416×240	3.055	1.471	14.178	0.163	4.664	**0.088**
832×480	11.341	1.677	14.924	0.171	4.759	**0.068**
1280×720	18.499	2.374	20.484	0.232	6.689	**0.090**
1920×1080	51.415	1.963	16.131	0.193	5.254	**0.076**
2560×1600	114.200	1.692	14.303	0.164	4.646	**0.067**
Average	-	1.835	16.004	0.185	5.202	**0.078**

more time overhead of an in-loop filter. Note that the results in this table are averaged over all JCT-VC test sequences with the same resolution. Considering that learning-based in-loop filters can be significantly accelerated by a GPU, both the results with and without GPU are provided, for our MIF approach and the RHCNN [20]. Here, the above learning-based approaches were implemented at the open-source machine learning framework TensorFlow (TM) [52]. We record the computational time of only using CPU and using CPU+GPU, respectively, for our approach and [20]. We can observe from Table VI that the heuristic approach [10] introduces the least time overhead among the three approaches, when implemented with only CPU. However, benefiting from the GPU acceleration, our MIF approach and the RHCNN-based in-loop filter can be drastically accelerated. As a result, our approach with GPU consumes the least time among all configurations in Table VI, which is 2.4 and 23.5 times faster than the RHCNN [20] with GPU and the heuristic loop filter [10] with only CPU, respectively. From the above analysis, the proposed MIF is an efficient approach in terms of time complexity, as a learning-based in-loop filter.

C. Analysis With Various Settings

Transfer to LDP configuration. In this section, we first evaluate the RD performance of our MIF approach at the LDP configuration through transfer learning, to verify its generalization ability. The models of MIF-Net and IF-Net at all four QPs $\{22, 27, 32, 37\}$ were initialized from those of the RA configuration at the corresponding QPs. Then, they were fine-tuned on our HIF database for the LDP configuration. The file *encoder_lowdelay_P_main.cfg* [49] was applied during both transfer learning and performance evaluation, while other experimental settings followed those at the RA configuration, as mentioned in Section VI-A. Table VII shows the RD performance of all four approaches at the LDP configuration. Note that the results are reported over all sequences at different classes/resolutions, from both the JCT-VT standard test set and our supplementary test set. We can observe from Table VII that our MIF approach achieves -23.341% of BD-BR on average, outperforming -16.567% of the DBF and SAO, -18.934% of [10] and -19.518% of [20]. Similar results can also be found in terms of BD-PSNR. In conclusion, the effectiveness and generalization ability of our MIF approach have been verified at the LDP configuration.

Statistics of frame quality. To better understand the performance of our MIF approach, it is helpful to analyze the statistics of frame quality, for different in-loop filters. In addition to PSNR, the structural similarity (SSIM) [53] is also added to evaluate the visual quality of video sequences. As tested in [53], SSIM has a remarkably better prediction of subjective visual quality than PSNR. Figure 12-(a) illustrates the quality fluctuation for the first 100 frames of sequence *KristenAndSara* as an example, evaluated on our MIF approach, standard HEVC and two state-of-the-art approaches [10], [20]. It can be observed that our MIF approach outperforms other three approaches in terms of overall PSNR and SSIM. Also, the quality fluctuation for our approach is less than that for other approaches. For more comprehensive analysis, we further compare the statistics of PSNR and SSIM for video sequences encoded by four approaches, as shown in Figure 12-(b). Here, the results are averaged over all 18 JCT-VC test sequences at the RA configuration with QPs $\{22, 27, 32, 37\}$. Analyzed from Figure 12-(b), our approach achieves the mean PSNR of 37.533 dB, considerably higher than that for other three approaches. Moreover, the standard deviation of PSNR for our approach is 0.776 dB, smaller than 0.791 dB of standard HEVC, 0.787 dB of [10] and 0.786 dB of [20]. For the mean and standard deviation of SSIM, similar results can be found. The above analysis verifies that our MIF

TABLE VII

RD PERFORMANCE OF IN-LOOP FILTERS (LDP CONFIG.)

Source	Class or Resolution	DBF and SAO		[10]		[20]		Proposed MIF	
		BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)
JCT-VC test set	Class A	-17.214	0.642	-20.979	0.813	-20.756	0.803	**-24.449**	**0.971**
	Class B	-20.225	0.515	-23.898	0.617	-22.944	0.593	**-26.065**	**0.677**
	Class C	-12.787	0.503	-13.698	0.540	-14.543	0.578	**-16.954**	**0.684**
	Class D	-8.399	0.361	-8.813	0.380	-9.827	0.429	**-11.888**	**0.517**
	Class E	-24.048	0.713	-27.577	0.824	-27.277	0.830	**-32.198**	**1.008**
Supple-mentary test set	352×288	-10.915	0.472	-11.995	0.523	-15.153	0.678	**-17.837**	**0.792**
	640×360	-11.626	0.559	-14.966	0.730	-13.985	0.683	**-17.970**	**0.885**
	720×480	-12.548	0.430	-14.029	0.486	-17.049	0.602	**-24.553**	**0.934**
	1280×720	-24.469	0.474	-26.205	0.515	-26.995	0.540	**-31.473**	**0.645**
	1920×1080	-23.438	0.758	-27.184	0.907	-26.653	0.861	**-30.021**	**0.986**
Average		-16.567	0.543	-18.934	0.634	-19.518	0.660	**-23.341**	**0.810**

approach can achieve both better overall quality and lower fluctuation of quality for compressed videos. This benefits from the multi-frame design, in which low-quality frames can be significantly enhanced using other higher-quality frames.

Comparison with learning-based in-loop filters on JVET. Considering that learning-based in-loop filters proposed by JVET have made remarkable achievements, it is also necessary to evaluate their performance. To this end, we compare our MIF approach with two sequence-independent filters for VVC, i.e., the residual weight-sharing CNN [35] and the dense residual CNN [36]. Note that the filters [35], [36] were re-implemented in HM [43] for HEVC. For fair comparison, the models of [35], [36] were both re-trained on our HIF database. Table VIII shows the RD performance of three approaches at the RA configuration with QPs $\{22, 27, 32, 37\}$. We can find in this table that the average BD-BR of our approach is -12.184%, outperforming -7.875% of [35] and -9.004% of [36]. In terms of BD-PSNR, there exist similar results. On a closer observation, our approach also performs better than the other two approaches at each resolution of both test sets in Table VIII, in terms of both BD-BR and BD-PSNR. Therefore, the effectiveness and stability of our MIF approach have been verified, compared with two newly-developed in-loop filter approaches in JVET.

TABLE VIII

RD PERFORMANCE OF THE PROPOSED FILTER AND FILTERS FOR JVET (RA CONFIG.)

Type	[35]		[36]		Proposed MIF	
	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)
$j1$	-10.34	0.412	-11.470	0.459	**-14.186**	**0.575**
$j2$	-9.835	0.226	-10.818	0.249	**-13.486**	**0.316**
$j3$	-5.647	0.219	-6.564	0.257	**-9.075**	**0.362**
$j4$	-3.683	0.168	-4.722	0.214	**-7.281**	**0.330**
$j5$	-10.450	0.342	-11.710	0.383	**-15.262**	**0.506**
$s1$	-7.724	0.362	-8.860	0.417	**-12.004**	**0.575**
$s2$	-7.222	0.345	-8.464	0.405	**-11.607**	**0.570**
$s3$	-6.719	0.277	-7.855	0.325	**-13.566**	**0.569**
$s4$	-9.153	0.189	-10.294	0.215	**-13.023**	**0.280**
$s5$	-7.979	0.266	-9.285	0.311	**-12.349**	**0.429**
Aver.	-7.875	0.281	-9.004	0.324	**-12.184**	**0.451**

Types for JCT-VC test set: $j1$: Class A, $j2$: Class B, $j3$: Class C, $j4$: Class D, $j5$: Class E.
Types for Supplementary test set: $s1$: 352×288, $s2$: 640×360, $s3$: 720×480, $s4$: 1280×720, $s5$: 1920×1080.

D. Ablation Study

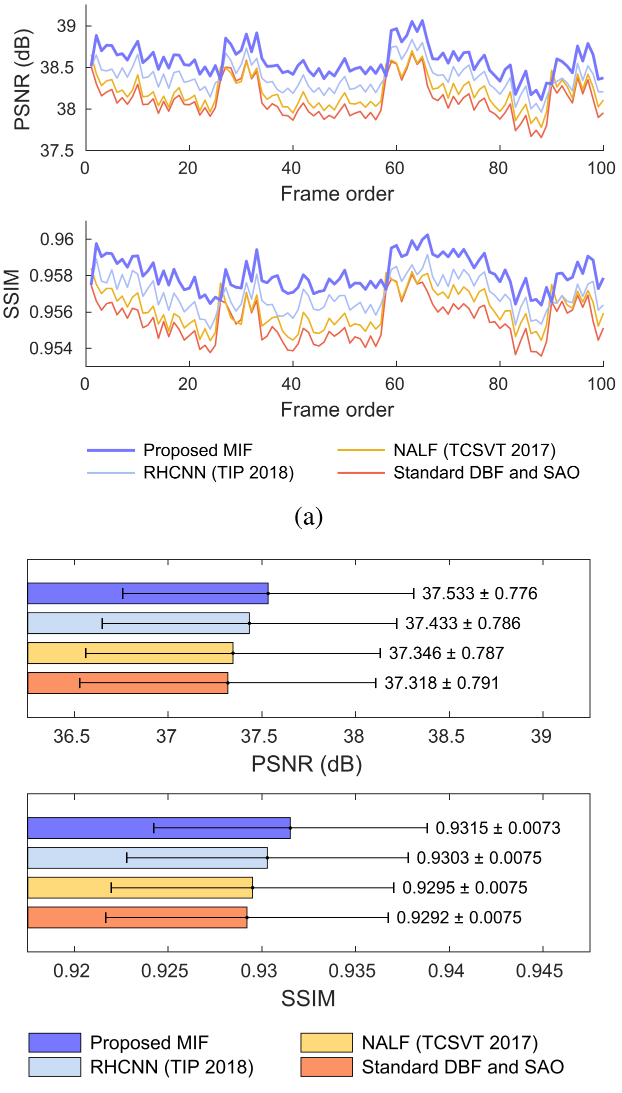
We further conduct a series of ablation experiments to investigate the effectiveness of major components in our approach. Our ablation study starts from the standard in-loop filter, and then certain components are added step-wise, finally reaching the proposed MIF approach. Figure 13 shows the RD performance at the RA configuration. More details are discussed in the following.

Plain CNN *vs.* standard in-loop filter. In the standard DBF and SAO, the filtering procedure is predetermined without any trainable parameter, which tends to be limited in reducing compression artifacts with diverse content. By contrast, we replace the standard DBF and SAO by a plain-CNN-based filter, providing sufficient trainable parameters. For simplicity, the plain CNN is composed of 4 successive convolutional layers[4] for generating the difference frame $\mathbf{F}_n^{\Delta}$, without any dense unit, block-adaptive convolutional layer and utilization of multiple frames. We can observe from Figure 13 that the plain CNN improves RD performance by 4.381% with BD-BR saving and 0.147 dB of BD-PSNR increase, compared with the standard DBF and SAO.

CNN with dense units *vs.* plain CNN. To analyze the impact of network topology, we substitute the 4 convolutional layers in the plain CNN by 4 successive dense units, i.e., changing the plain CNN into the proposed IF-Net without block-adaptive convolution. Note that the numbers of trainable parameters are the same in both networks with and without dense units, each containing 47, 196 convolutional weights. As can be seen in Figure 13, the dense units outperform the plain typology of CNN by 0.891% of BD-BR saving and 0.032 dB of BD-PSNR increase, when implemented in the proposed in-loop filter. As discussed in [28], some possible reasons for the effectiveness of dense units include: (1) the feature reuse to enhance parameter efficiency, (2) the flexible topology with various depth of pathways for easy convergence and (3) the implicit deep supervision that enforces intermediate layers to learn discriminative features.

IF-Net with *vs.* without block-adaptive convolution. In HEVC, the flexible CTU partition structure has evident influence on compression artifacts, especially on blocking artifacts. Therefore, a block-adaptive convolutional layer is proposed to handle such artifacts. We evaluate two networks

[4] The plain CNN contains 3 layers (each outputting 48 channels) and 1 layer (outputting 1 channel) in sequence. All layers are convoluted by 3×3 kernels with stride of 1, followed by the PReLU [45] activation.



(a)

(b)

Fig. 12. Statistics of frame quality for different approaches. (a) PSNR and SSIM fluctuation for the first 100 frames of sequence *KristenAndSara*. (b) Statistics of PSNR and SSIM over all 18 JCV-VC sequences at the RA configuration with QPs {22, 27, 32, 37}. "$a \pm b$" represents mean value of a with standard deviation of b.

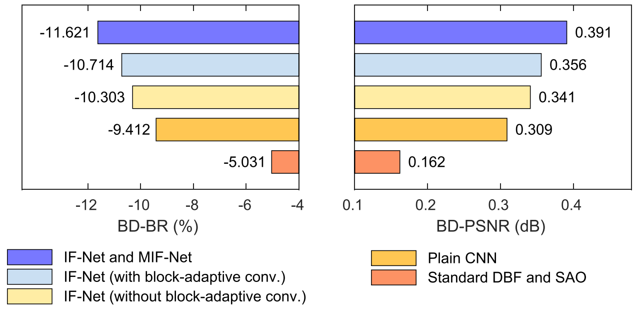


Fig. 13. RD performance of ablation study. The results are obtained over all 18 sequences in the JCT-VT test set, compared at the RA configuration with QPs {22, 27, 32, 37}.

with and without block-adaptive convolution to analyze its effectiveness. In terms of RD performance, the IF-Net with block-adaptive convolution is better than that without it, where the decrease of BD-BR is 0.411% and the increase of PSNR is 0.015 dB. The block-adaptive convolution outperforms typical convolution, because the CTU structure has impact on the compression artifacts in HEVC, especially the blocking artifacts.

IF-Net and MIF-Net *vs.* only IF-Net. In our MIF approach, the utilization of multiple frames is a major contribution for enhancing the quality of each URF. Here, we investigate the RD performance of our approach with and without MIF-Net. Note that RFS is enabled when evaluating the performance with MIF-Net. As can be seen in Figure 13, our approach with both IF-Net and MIF-Net outperforms the setting with only IF-Net (0.907% BD-BR reduction and 0.035 dB BD-PSNR increase). These results verify the effectiveness of leveraging multiple frames for our in-loop filter. The multi-frame design is effective because there always exists considerable quality fluctuation among adjacent frames, and a low-quality frame can be enhanced by its neighboring higher-quality frames.

From the above analysis, three major configurations of network contribute to the proposed MIF approach, comparing with a plain CNN. Among them, the outperformance is mainly due to the utilization of multiple frames and the efficient dense units. Also, the novel block-adaptive convolution helps to improve the RD performance in a certain extent. Therefore, the reason why such network configurations are beneficial, lies in both the advanced topology itself (i.e., the dense units) and the specific characteristics of the compression artifacts (i.e., the multi-frame design and block-adaptive convolution).

VII. Conclusion

In this paper, we have proposed a deep-learning-based MIF approach for HEVC. Different from the existing in-loop filter approaches based on a single frame, our MIF approach learns to enhance the visual quality of one frame by leveraging multiple adjacent frames. To this end, we first constructed a large-scale HIF database, and found that there normally exist an adequate number of reference frames with both higher quality and similar content for a URF. According to our observation, we design an RFS for selecting these reference frames. Taking advantage of the HIF database, a deep MIF-Net model was proposed to enhance the quality of each URF, which utilizes both the spatial information of this URF and the temporal information of its selected reference frames. The MIF-Net model was constructed by the newly developed DenseNet with improved generalization ability and computational efficiency. Also, a novel block-adaptive convolutional layer was proposed for MIF-Net, considering the blocking artifacts highly influenced by the CTU structure in HEVC. Finally, both objective and subjective experiments demonstrated that our MIF approach significantly outperforms the standard in-loop filter and other state-of-the-art approaches for HEVC. For future works, more various details related to compression artifacts (e.g., skip modes, prediction unit partition, motion vectors and residual frames) may also be utilized, with potential to further improve the performance of in-loop filters. In addition, the implementation of deep neural networks can be accelerated with some techniques [54]. Thus, another future work is applying these techniques to speed up our MIF approach.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] M. Yuen and H. R. Wu, "A survey of hybrid MC/DPCM/DCT video coding distortions," *Signal Process.*, vol. 70, no. 3, pp. 247–278, Nov. 1998. doi: 10.1016/S0165-1684(98)00128-5.

[4] A. Norkin *et al.*, "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.

[5] C.-M. Fu *et al.*, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[6] C.-Y. Tsai *et al.*, "Adaptive loop filtering for video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 934–945, Dec. 2013.

[7] M. Mtsumura, Y. Bandoh, S. Takamura, and H. Jozawa, *In-Loop Filter Based on Non-Local Means Filter, Joint Collaborative Team on Video Coding*, document Rec. JCTVC-E206, ITU-T SG16, Geneva, Switzerland, Mar. 2011.

[8] Q. Han, R. Zhang, W.-K. Cham, and Y. Liu, "Quadtree-based non-local Kuan's filtering in video compression," *J. Visual Commun. Image Represent.*, vol. 25, no. 5, pp. 1044–1055, Jul. 2014.

[9] S. Ma, X. Zhang, J. Zhang, C. Jia, S. Wang, and W. Gao, "Nonlocal in-loop filter: The way toward next-generation video coding?" *IEEE Multimedia*, vol. 23, no. 2, pp. 16–26, Apr./Jun. 2016.

[10] X. Zhang *et al.*, "Low-rank-based nonlocal adaptive loop filter for high-efficiency video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2177–2188, Oct. 2017.

[11] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: A deep learning approach," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5044–5059, Oct. 2018.

[12] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for HEVC compressed videos," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[13] J. Lu, V. E. Liong, and J. Zhou, "Deep hashing for scalable image search," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2352–2367, May 2017.

[14] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou, "Learning deep binary descriptor with multi-quantization," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[15] W. S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *Proc. IEEE IVMSP Workshop*, Jul. 2016, pp. 1–5.

[16] Y. Dai, D. Liu and F. Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," in *Proc. MMM*, Reykjavik, Iceland, Jan. 2017, pp. 28–39.

[17] J. Kang, S. Kim, and K. M. Lee, "Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec," in *Proc. IEEE ICIP*, Sep. 2017, pp. 26–30.

[18] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

[19] X. Meng, C. Chen, S. Zhu, and B. Zeng, "A new HEVC in-loop filter based on multi-channel long-short-term dependency residual networks," in *Proc. DCC*, Mar. 2018, pp. 187–196.

[20] 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 Trans. Image Process.*, vol. 27, no. 8, pp. 3827–3841, Aug. 2018.

[21] F. Brandi, R. de Queiroz, and D. Mukherjee, "Super-resolution of video using key frames and motion estimation," in *Proc. IEEE ICIP*, Oct. 2008, pp. 321–324.

[22] B. C. Song, S.-C. Jeong, and Y. Choi, "Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 3, pp. 274–285, Mar. 2011.

[23] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Trans. Comput. Imaging*, vol. 2, no. 2, pp. 109–122, Jun. 2016.

[24] D. Li and Z. Wang, "Video superresolution via motion compensation and deep residual learning," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 749–762, Dec. 2017.

[25] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1015–1028, Apr. 2018.

[26] J. Caballero *et al.*, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. IEEE CVPR*, Jul. 2017, pp. 2848–2857.

[27] O. Makansi, E. Ilg, and T. Brox, "End-to-end learning of video super-resolution with motion compensation," in *Proc. GCPR*, Aug. 2017, pp. 203–214.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, Jul. 2017, pp. 4700–4708.

[29] T. Li, M. Xu, R. Yang, and X. Tao, "A DenseNet based approach for multi-frame in-loop filter in HEVC," in *Proc. DCC*, Mar. 2019, pp. 270–279.

[30] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 184–199.

[31] Y. L. Hsiao, C. Y. Chen, T. D. Chuang, C. W. Hsu, Y. W. Huang, and S. M. Lei, *Convolutional Neural Network Loop Filter*, document Rep. JVET-M0159, Marrakech, MA, USA, Jan. 2019.

[32] H. Yin, R. Yang, X. Fang, S. Ma, and Y. Yu, *Adaptive Convolutional Neural Network Loop Filter*, document Rep. JVET-M0566, Marrakech, MA, USA, Jan. 2019.

[33] K. Kawamura and S. Naito, *A Result of Convolutional Neural Network Filter*, document Rep. JVET-M0872, Marrakech, MA, USA, Jan. 2019.

[34] C. Lin, J. Yao, and L. Wang, *Convolutional Neural Network Filter (CNNF) for Intra Frame*, document Rep. JVET-M0351, Marrakech, MA, USA, Jan. 2019.

[35] Y. Dai, D. Liu, Y. Li, and F. Wu, *CNN-Based in-Loop Filter Proposed by USTC*, document Rep. JVET-M0510, Marrakech, MA, USA, Jan. 2019.

[36] Y. Wang, Z. Chen, Y. Li, L. Zhao, S. Liu, and X. Li, *Test Results of Dense Residual Convolutional Neural Network Based in-Loop Filter*, document Rep. JVET-M0508, Marrakech, MA, USA, Jan. 2019.

[37] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proc. IEEE CVPR*, Jun. 2018, pp. 6664–6673.

[38] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 3, pp. 475–489, Jun. 2014.

[39] Xiph. (2017). *Xiph. Org Video Test Media*. [Online]. Available: https://media.xiph.org/video/derf

[40] CDVL. (2019). *Consumer Digital Video Library*. [Online]. Available: https://www.cdvl.org/index.php

[41] VQEG. (2017). *VQEG Video Datasets and Organizations*. [Online]. Available: https://www.its.bldrdoc.gov/vqeg/video-datasets-and-organizations.aspx/%

[42] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[43] JCT-VC. (2014). *HM Software*. Accessed: Nov. 5, 2016. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.5/

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, May 2015, pp. 1–15.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.

[47] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. IEEE IWAENC*, vol. 15, Jun. 2011, pp. 315–323.

[48] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE CVPR*, Jun. 2016, pp. 4724–4732.

[49] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document Rep. JCTVC-L1100, Jan. 2013.

[50] G. Bjøntegaard, *Calculation of Average PSNR Difference Between RD-Curves*, document Rep. ITU-T-VCEG-M33, Austin, TX, USA, Apr. 2001.

[51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, vol. 9, Mar. 2010, pp. 249–256.

[52] Google. (2018). *Tensorflow*. Accessed: Mar. 23, 2019. [Online]. Available: https://www.tensorflow.org/

[53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[54] S. Han, H. Mao, and B. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. ICLR*, May 2016, pp. 1–14.