# A DenseNet Based Approach for Multi-Frame In-Loop Filter in HEVC

Tianyi Li*, Mai Xu*, Ren Yang* and Xiaoming Tao†

*School of Electronic and Information Engineering,
Beihang University, Beijing, 100191, China
† Department of Electronic Engineering,
Tsinghua University, Beijing, 100084, China
maixu@buaa.edu.cn (Corresponding author: Mai Xu)

## Abstract

High efficiency video coding (HEVC) has brought outperforming efficiency for video compression. To reduce the compression artifacts of HEVC, we propose a DenseNet based approach as the in-loop filter of HEVC, which leverages multiple adjacent frames to enhance the quality of each encoded frame. Specifically, the higher-quality frames are found by a reference frame selector (RFS). Then, a deep neural network for multi-frame in-loop filter (named 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 the improved generalization capacity and computational efficiency. Finally, experimental results verify the effectiveness of our multi-frame in-loop filter, outperforming the HM baseline and other state-of-the-art approaches.

## 1 Introduction

The high efficiency video coding (HEVC) standard [1] developed by the Joint Collaborate Team on Video Coding (JCT-VC) has brought outperforming efficiency for video compression. However, various artifacts (e.g., blocking, blurring and ringing artifacts) also exist in compressed videos, mainly resulting from the block-wise prediction and quantization with limited precision. To alleviate such artifacts, in-loop filters were adopted for 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 built-in in-loop filters were proposed for standard HEVC, including deblocking filter (DBF) [2], sample adaptive offset (SAO) filter [3] and adaptive loop filter (ALF) [4]. Specifically, DBF is firstly used to remove the blocking artifacts. Then, the SAO filter reduces distortion by adding an adaptive offset to each sample. Afterwards, ALF minimizes the distortion based on Wiener filter. However, ALF introduces heavy bit-rate overhead and it has not been adopted in the final version of HEVC. Besides the built-in in-loop filters for HEVC, various heuristic and learning-based methods have also been proposed. In heuristic methods, some prior knowledge of video coding is utilized to build a statistical model of compression artifacts, and then each frame is enhanced based on the model. For example, Matsumura *et al.* [5] utilized the weighted mean of non-local similar frame patches for artifact reduction.

2375-0359/19/$31.00 ©2019 IEEE
DOI 10.1109/DCC.2019.00035

Zhang *et al.* [6] attached a low-rank constraint on each matrix formed by a patch group, and then established an adaptive soft-thresholding model to achieve sparse representation. More recently, deep learning has been successfully employed in many areas about data compression, such as video coding [7] and quality enhancement [8]. Also, learning-based methods have further improved the performance of in-loop filtering. Among them, Meng *et al.* [9] developed a multi-channel long-short-term dependency residual network (MLSDRN) for mapping a distorted frame to the raw frame, inserted between DBF and SAO. Zhang *et al.* [10] proposed a residual highway CNN (RHCNN) based on the ResNet [11], implemented after the standard SAO. However, none of the above learning-based methods has employed multiple frames for in-loop filtering in HEVC. Typically, the high fluctuation of visual quality exists across the encoded frames, and thus a low-quality frame can be enhanced by referring to its adjacent higher-quality frames.

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 exploit the quality fluctuation of encoded frames via designing a reference frame selector (RFS) to find reference frames for an unfiltered reconstructed frame (URF), 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 multiples frames. In the case that no sufficient reference frames are selected by RFS, a simpler deep neural network for in-loop filter (named IF-Net) is used to enhance the URF instead. Considering the blocking artifacts influenced by the coding tree unit (CTU) partition, the proposed networks are also adaptive to the partition structure, via varying convolutional kernels at different locations of the coding unit (CU) and transform unit (TU) maps. Finally, the experimental results show that our approach outperforms other state-of-the-art approaches, with 5.33% and 2.40% saving of the Bjøntegaard delta bit-rate (BD-BR) over the non-local adaptive loop filter [6] and the RHCNN [10], respectively.

## 2 Proposed MIF Approach

### *2.1 Framework*

The framework of our MIF approach is illustrated in Figure 1. 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 2.2. Next, one of the two possible filtering modes is adopted 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. In MIF-Net, each reference frame is first aligned with the URF in terms of content, with a motion compensation network. Then,

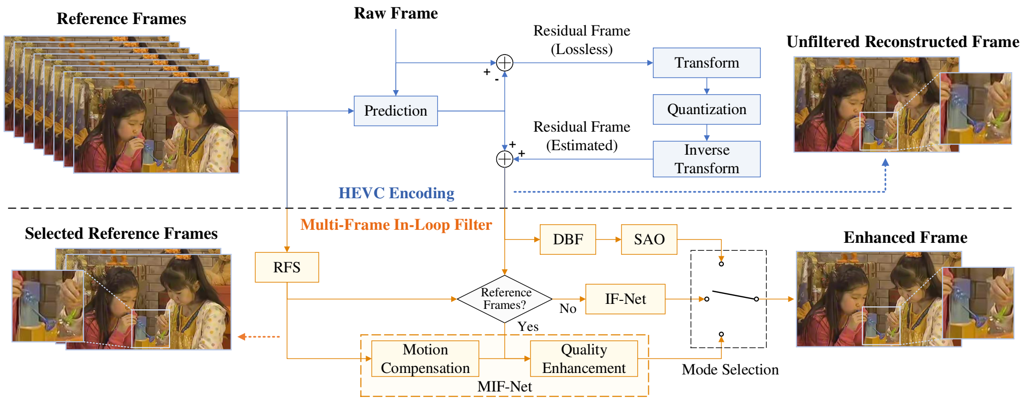

Figure 1: Framework of the proposed MIF.

both aligned reference frames and the URF are fed into a quality enhancement network to output the reconstructed frame.

- **Mode 2: IF-Net.** If no enough reference frames are found for the URF, IF-Net is adopted instead for quality enhancement. In contrast to MIF-Net, IF-Net takes only the URF as input without any consideration of multiple frames.

More details about Modes 1 and 2 are presented in Section 2.3. If MIF-Net or 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.

### *2.2 Design of RFS*

In our approach, RFS selects reference frames for each URF. For the $n$-th URF (denoted as $\mathbf{F}_n^{\mathrm{U}}$) in a 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.

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

Based on the above metrics, the reference frame pool is first divided into valid and invalid reference frames, and then all valid frames are fed into RFS-Net to select $M$ frames in final. Specifically, 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 increment is positive and the CC value is above a threshold $\tau$, i.e., $V_{i,n} = 1$ in (1), $\mathbf{F}_i^{\mathrm{P}}$ is seen as a valid reference frame.

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

If there exist at least $M$ valid reference frames, the six metrics for each valid reference frame form a 6-dimensional vector, and then they are input to a two-layer

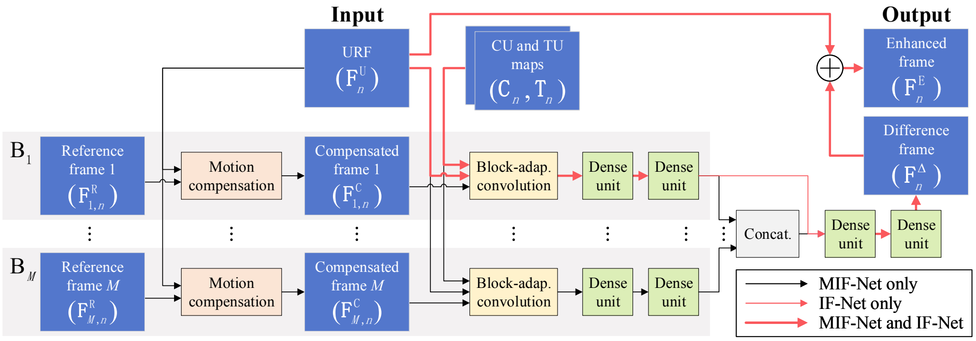

Figure 2: Architecture of MIF-Net or IF-Net.

fully connected network (named RFS-Net[1]) to output a scalar $\hat{R}_{i,n}$. The output $\hat{R}_{i,n}$ is a continuous variable representing the potential of $\mathbf{F}_i^{\mathrm{P}}$ being the reference for $\mathbf{F}_n^{\mathrm{U}}$. A larger $\hat{R}_{i,n}$ indicates that $\mathbf{F}_i^{\mathrm{P}}$ has more potential than other reference frames for enhancing $\mathbf{F}_n^{\mathrm{U}}$. Here, $\hat{R}_{i,n}$ is the predicted value by RFS-Net, with the corresponding ground-truth denoted by $R_{i,n}$. In RFS-Net, the ground-truth $R_{i,n}$ should reflect the quality of a valid reference frame after it is aligned with $\mathbf{F}_n^{\mathrm{U}}$ via motion compensation. To this end, we assign $R_{i,n}$ as the PSNR between the compensated valid reference frame and the $n$-th raw frame (denoted as $\mathbf{F}_n$). In accord with $\hat{R}_{i,n}$, the $R_{i,n}$ is also Z-scored normalized within one training batch. After normalization, the $\ell_2$-loss on the whole training batch can be used to measure the difference between $R_{i,n}$ and $\hat{R}_{i,n}$, formulated as

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

which is optimized by the Adam algorithm [13]. Using the trained RFS-Net model, the reference potential for all the valid frames can be obtained. Then, RFS selects $M$ frames 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.

### *2.3 MIF-Net and IF-Net*

This section mainly focuses on the architecture of MIF-Net and its training strategy, and then specifies the difference between IF-Net and MIF-Net. Figure 2 illustrates the overall architecture of MIF-Net or 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. The information from $M$ parallel branches $\{\mathbf{B}_m\}_{m=1}^{M}$ is synthesized, with each branch $\mathbf{B}_m$ dealing with the corresponding reference frame $\mathbf{F}_{m,n}^{\mathrm{R}}$. In branch $\mathbf{B}_m$, $\mathbf{F}_{m,n}^{\mathrm{R}}$ is first aligned with $\mathbf{F}_n^{\mathrm{U}}$ to produce a motion-compensated frame, denoted as $\mathbf{F}_{m,n}^{\mathrm{C}}$. Next, $\mathbf{F}_n^{\mathrm{U}}$ with $\mathbf{F}_{m,n}^{\mathrm{C}}$ flows through a novel convolutional layer guided by the CTU partition structure of $\mathbf{F}_n^{\mathrm{U}}$ (named block-adaptive convolutional

[1] The 6-dimensional vector flows through two layers, with 12 hidden nodes and 1 output node, respectively. Both layers are activated with parametric rectified linear units (PReLU) [12]. Note that the samples in one training batch are extracted from the valid reference frames for only one URF, and the output of samples in the same batch are Z-score normalized.

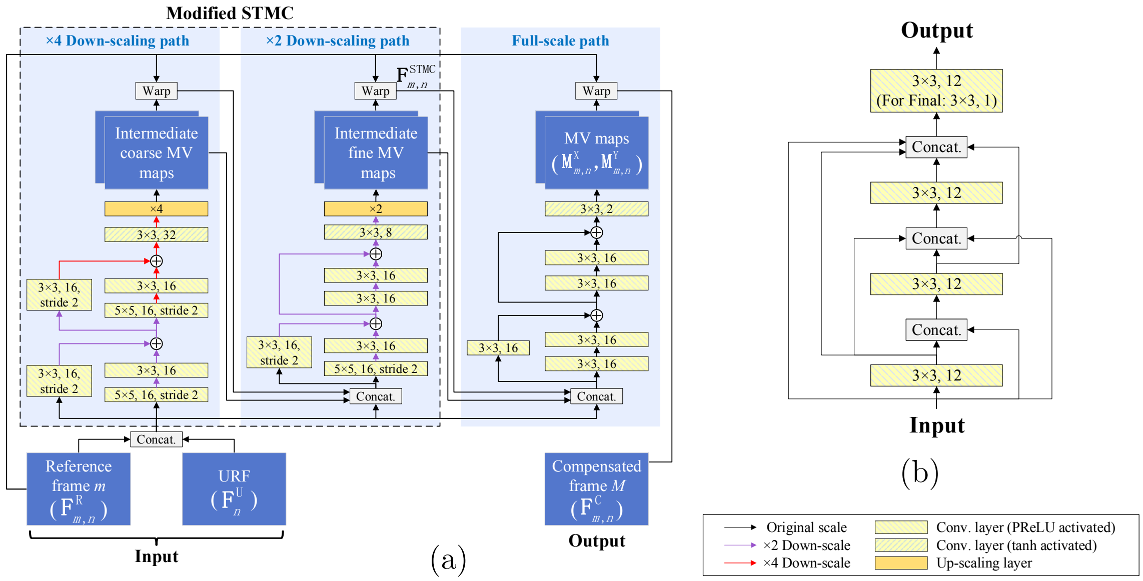

Figure 3: Network details. (a) Motion compensation network. (b) Dense unit. For convolutional layers, "$p \times p$, $q$" represents $q$ output channels with $p \times p$ kernels. Note that the convolutional stride is set to 1 by default, except that explicitly mentioned in certain layers.

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 dense units [14] to extract more comprehensive features within $\mathbf{B}_m$. Finally, the extracted features from $M$ 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 as $\mathbf{F}^{\Delta}_n$) is regarded as a difference frame, and the enhanced frame $\mathbf{F}^{\mathrm{E}}_n$ is the summation of $\mathbf{F}^{\Delta}_n$ and $\mathbf{F}^{\mathrm{U}}_n$. The details of MIF-Net components are presented in the following.

**Motion compensation network.** We propose a motion compensation network based on the spatial transformer motion compensation (STMC) [15], for content alignment between $\mathbf{F}^{\mathrm{R}}_{m,n}$ and $\mathbf{F}^{\mathrm{U}}_n$, illustrated in Figure 3-(a). In [15], the STMC takes both $\mathbf{F}^{\mathrm{R}}_{m,n}$ and $\mathbf{F}^{\mathrm{U}}_n$ as the input, to output a compensated frame denoted as $\mathbf{F}^{\mathrm{STMC}}_{m,n}$. The STMC consists of two paths (×4 and ×2 down-scaling paths) to predict different precision of motion vector (MV) maps, and the MV maps from the ×2 down-scaling path are applied to $\mathbf{F}^{\mathrm{R}}_{m,n}$ for outputting $\mathbf{F}^{\mathrm{STMC}}_{m,n}$. The two down-sampling paths in [15] are capable for estimating various scales of motion. However, the accuracy of the STMC is limited due to down-sampling, and its architecture can also be improved. Therefore, we propose a motion compensation network with the following advancements. (1) Besides the ×2 and ×4 down-scaling paths, a full-scale path is added to enhance the precision of MV estimation; (2) Inspired by the ResNet [11], 6 shortcuts are added next to the convolutional layers for higher network capacity and ease to be trained; (3) All ReLU [16] activation for convolutional layers are replaced by PReLU [12]. With the above advancements, 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

$$\mathbf{F}^{\mathrm{C}}_{m,n}(x,y) = \mathrm{Bil}\{\mathbf{F}^{\mathrm{R}}_{m,n}(x + \mathbf{M}^{\mathrm{X}}_{m,n}(x,y), y + \mathbf{M}^{\mathrm{Y}}_{m,n}(x,y))\}, \tag{3}$$

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 layer.** The input to this layer is a concatenation of three feature maps, including a compensated frame $\mathbf{F}^{\mathrm{C}}_{m,n}$, a URF $\mathbf{F}^{\mathrm{U}}_{n}$ and $\mathbf{F}^{\mathrm{C}}_{m,n}-\mathbf{F}^{\mathrm{U}}_{n}$. The CU and TU partition are represented by two feature maps, i.e., $\mathbf{C}_n$ and $\mathbf{T}_n$, respectively. $\mathbf{C}_n$ and $\mathbf{T}_n$ each has the same size as $\mathbf{F}^{\mathrm{U}}_{n}$, and the values in the two maps are assigned according to the partition structure. If pixel $(x, y)$ is on the boundary of a CU or TU, $\mathbf{C}_n(x, y)$ or $\mathbf{T}_n(x, y)$ is set to 1. Otherwise, the value is set to $-1$. Afterwards, the target of this 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, assuming that $P^{\mathrm{I}}$, $P^{\mathrm{G}}$ and $P^{\mathrm{O}}$ feature maps are used as the input, guidance and output, respectively. The guided convolution consists of two main procedures, i.e., intermediate map generation and convolution with intermediation. First, the $P^{\mathrm{G}}$ guidance feature maps are processed with two typical convolutional layers to generate $P^{\mathrm{O}}$ intermediate feature maps, keeping the size of each feature map unchanged. Then, during the convolution, the $P^{\mathrm{O}}$ output feature maps are generated based on these $P^{\mathrm{O}}$ intermediate feature maps, correspondingly. Compared with typical convolution using space-irrelevant weights $w_{j,l}$ only, the guided convolution is conducted with space-relevant weights $w^{\mathrm{G}}_{j,l}$ generated from the intermediation, as formulated below

$$w^{\mathrm{G}}_{j,l}(\Delta x, \Delta y) = w_{j,l}(\Delta x, \Delta y) \cdot \mathbf{F}^{\mathrm{M}}_{l}(x + \Delta x, y + \Delta y), \tag{4}$$

$$\mathbf{F}^{\mathrm{O}}_{l}(x, y) = \sum_{j=1}^{P^{\mathrm{I}}} \sum_{\Delta x=-1}^{1} \sum_{\Delta y=-1}^{1} w^{\mathrm{G}}_{j,l}(\Delta x, \Delta y) \cdot \mathbf{F}^{\mathrm{I}}_{j}(x + \Delta x, y + \Delta y). \tag{5}$$

In (4) and (5), $\mathbf{F}^{\mathrm{I}}_{j}$, $\mathbf{F}^{\mathrm{M}}_{l}$ and $\mathbf{F}^{\mathrm{O}}_{l}$ represent the $j$-th input, the $l$-th intermediate and the $l$-th output feature maps, respectively. $\Delta x, \Delta y$ denote the relative coordinates within a $3 \times 3$ kernel. For each block-adaptive convolutional layer in MIF-Net, there exist $P^{\mathrm{I}} = 3$ and $P^{\mathrm{G}} = 2$, and we set the number of output maps to be $P^{\mathrm{O}} = 16$.

**Dense units for quality enhancement.** The DenseNet [14] introduces various length of inter-layer connections, with alleviation of vanishing gradients and encouragement of feature reuse. Considering the advantages, $(2M + 2)$ dense units are adopted in MIF-Net, i.e., 2 dense units in each branch and 2 dense units at the end of MIF-Net synthesizing features from $M$ branches. Figure 3-(b) illustrates the structure of each dense unit, and it can be observed that a dense unit with 4 convolutional layers includes 10 inter-layer connections, much more than a 4-layer plain CNN with only 4 inter-layer connections. Here, each layer outputs 12 channels, except the last layer in the final dense unit outputting only 1 channel as the difference frame $\mathbf{F}^{\Delta}_{n}$.

**MIF-Net Training.** With both motion compensation and quality enhancement, it may be difficult to train the whole MIF-Net directly. Thus, we propose to train it with intermediate supervision [17], introducing two loss functions at different stages. First, the difference between $\mathbf{F}^{\mathrm{U}}_{n}$ and each frame in $\{\mathbf{F}^{\mathrm{C}}_{m,n}\}_{m=1}^{M}$ can measure the performance of motion compensation, and thus it is defined as the intermediate loss

$$L_{\mathrm{INT}} = \frac{1}{M} \sum_{m=1}^{M} \|\mathbf{F}^{\mathrm{C}}_{m,n} - \mathbf{F}^{\mathrm{U}}_{n}\|_2^2, \tag{6}$$

where $\|\cdot\|_2$ represents the $\ell_2$-norm difference. Next, the difference between $\mathbf{F}_n^{\mathrm{E}}$ and $\mathbf{F}_n$ indicates the performance of the whole MIF-Net, and thus the global loss is

$$L_{\mathrm{GLO}} = \|\mathbf{F}_n^{\mathrm{E}} - \mathbf{F}_n\|_2^2. \tag{7}$$

The loss for training MIF-Net is the weighted summation of them:

$$L = \alpha \cdot L_{\mathrm{INT}} + \beta \cdot L_{\mathrm{GLO}}, \tag{8}$$

where $\alpha$ and $\beta$ are adjustable positive weights. On account that quality enhancement relies on the well-trained motion compensation network, $L_{\mathrm{INT}}$ should be emphatically optimized with $\alpha \gg \beta$ at early stage of training. After $L_{\mathrm{INT}}$ converges, we set $\beta \gg \alpha$ instead, to emphasize more on optimization of the global loss $L_{\mathrm{GLO}}$.

**Difference between IF-Net and MIF-Net.** The difference between two networks lies in the absence of $M$ reference frames in IF-Net. Therefore, only quality enhancement without motion compensation is adopted in IF-Net, illustrated by red arrows in Figure 2. Compared with MIF-Net, only one branch without any compensated frame exists in IF-Net, and the concatenation synthesizing $M$ branches is also omitted. Despite simpleness, a block-adaptive convolutional layer and four consecutive dense units still exist in IF-Net, ensuring sufficient network capacity. Considering no motion compensation in IF-Net, the loss of IF-Net is the same as $L_{\mathrm{GLO}}$ in MIF-Net.

## 3 Experimental Results

### *3.1 Settings*

**Experimental configurations.** In the experiments, all approaches for in-loop filtering were incorporated into the HEVC reference software HM 16.5. For our MIF approach, we established a large-scale database for HEVC in-loop filtering (named HIF database) containing 111 raw video sequences, collected from the JCT-VC [18], Xiph.org [19] and the conversational video set [20]. Our HIF database was divided into non-overlapping sets of training (83 sequences), validation (10 sequences) and test (18 sequences). The training set was used to train the networks, and the hyper-parameters in our approach were tunned on the validation set. The test set was used for performance evaluation, containing all 18 standard sequences from the JCT-VC set [18]. The RA configuration was applied for both network training and performance evaluation at four QPs, $\{22, 27, 32, 37\}$. During evaluation, the BD-BR and the Bjøntegaard delta PSNR (BD-PSNR) were measured to assess the rate-distortion (RD) performance.

**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 1. For training MIF-Net and IF-Net, all the frames were segmented into 64×64 patches. Here, each training sample was composed of the co-located patches from a raw frame, a URF, a CU map, a TU map and $M$ reference frames (if have). Considering the efficiency of training, the IF-Net or MIF-Net model at QP = 37 was trained from scratch, while the models at QPs $\{22, 27, 32\}$ were fine-tuned from the trained models at QPs $\{27, 32, 37\}$, respectively.

Table 1: Hyper-parameters for networks

| Hyper-parameter | RFS-Net | MIF-Net or IF-Net |
|---|---|---|
| Size of ref. frame pool: $N$ | 16 | - |
| Threshold for CC value: $\tau$ | 0.3 | - |
| Num. of selected ref. frames: $M$ | 2 | |
| Optimization | Adam algorithm [13] | |
| Batch size | $\leq 16^*$ | 16 |
| Initial learning rate | $10^{-5}$ | $10^{-4}$ |
| Num. of iterations | $10^5$ | $10^6$ (from scratch) or $2 \times 10^5$ (fine-tunning) |
| Changeable weights<br>in MIF-Net: $\alpha$ and $\beta$ | - | 0.99 & 0.01 (at beginning)<br>0.01 & 0.99 (after $L_{\text{INT}}$ converged) |

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

Table 2: RD performance of in-loop filters on the JCT-VC test set

| Class | Sequence | Standard DBF and SAO | | Non-local adaptive loop filter [6] | | RHCNN [10] | | Proposed MIF | |
|---|---|---|---|---|---|---|---|---|---|
| | | BD-BR (%) | BD-PSNR (dB) | BD-BR (%) | BD-PSNR (dB) | BD-BR (%) | BD-PSNR (dB) | BD-BR (%) | BD-PSNR (dB) |
| A | *PeopleOnStreet* | -8.29 | 0.37 | -12.03 | 0.54 | -12.48 | 0.57 | **-16.82** | **0.78** |
| | *Traffic* | -5.35 | 0.16 | -6.17 | 0.19 | -9.81 | 0.30 | **-12.15** | **0.38** |
| B | *BasketballDrive* | -6.65 | 0.15 | -8.84 | 0.20 | -11.05 | 0.25 | **-14.87** | **0.35** |
| | *BQTerrace* | -7.15 | 0.11 | -11.40 | 0.17 | -14.36 | 0.23 | **-17.13** | **0.27** |
| | *Cactus* | -7.54 | 0.16 | -8.90 | 0.19 | -12.52 | 0.27 | **-15.83** | **0.35** |
| | *Kimono* | -7.54 | 0.22 | -9.26 | 0.27 | -10.48 | 0.31 | **-12.24** | **0.37** |
| | *ParkScene* | -3.68 | 0.11 | -4.08 | 0.12 | -5.94 | 0.18 | **-7.99** | **0.25** |
| C | *BasketballDrill* | -5.02 | 0.21 | -5.39 | 0.22 | -7.81 | 0.33 | **-10.32** | **0.43** |
| | *BQMall* | -3.93 | 0.15 | -4.45 | 0.17 | -7.65 | 0.30 | **-9.38** | **0.37** |
| | *PartyScene* | -1.05 | 0.04 | -1.22 | 0.05 | -2.41 | 0.10 | **-4.16** | **0.17** |
| | *RaceHorses* | -6.15 | 0.22 | -7.08 | 0.26 | -10.40 | 0.39 | **-12.74** | **0.48** |
| D | *BasketballPass* | -3.85 | 0.18 | -4.32 | 0.20 | -7.68 | 0.37 | **-9.98** | **0.48** |
| | *BlowingBubbles* | -0.83 | 0.03 | -0.83 | 0.03 | -3.05 | 0.12 | **-3.98** | **0.16** |
| | *BQSquare* | -0.05 | 0.00 | 0.01 | 0.00 | -3.24 | 0.12 | **-4.40** | **0.17** |
| | *RaceHorses* | -4.44 | 0.20 | -4.80 | 0.22 | -8.84 | 0.41 | **-10.99** | **0.51** |
| E | *FourPeople* | -7.02 | 0.26 | -8.49 | 0.32 | -13.92 | 0.54 | **-16.48** | **0.64** |
| | *Johnny* | -5.60 | 0.14 | -8.03 | 0.21 | -11.62 | 0.30 | **-14.37** | **0.38** |
| | *KristenAndSara* | -6.41 | 0.20 | -8.01 | 0.25 | -12.62 | 0.41 | **-15.34** | **0.50** |
| Average | | -5.03 | 0.16 | -6.29 | 0.20 | -9.22 | 0.30 | **-11.62** | **0.39** |

### *3.2 Performance Evaluation*

**Objective RD performance.** We analyze the objective performance of our MIF approach in terms of the BD-BR and BD-PSNR, compared with the standard in-loop filters (DBF and SAO), a model-based approach (the non-local adaptive loop filter [6]) and a deep-learning-based approach (the RHCNN [10]). For a fair comparison, the RHCNN models in [10] were re-trained on our HIF database. Table 2 tabulates the RD performance of all four approaches, and the original HM without in-loop filters is used as anchor. As indicated in Table 2, the BD-BR of our MIF approach is $-11.62\%$ averaged over the 18 standard test sequences, outperforming $-5.03\%$ of the HM baseline, $-6.29\%$ of [6] and $-9.22\%$ of [10]. In terms of BD-PSNR, our approach achieves 0.39dB for the standard test set, also significantly better than 0.16dB of the HM baseline, 0.20dB of [6] and 0.30dB of [10], respectively. Therefore, our MIF approach achieves the best RD performance among all four approaches. The advancement of our approach mainly attributes to the accurate mapping from a URF to its corresponding raw frame, benefiting from the deep MIF-Net and IF-Net learned on our large-scale HIF database.

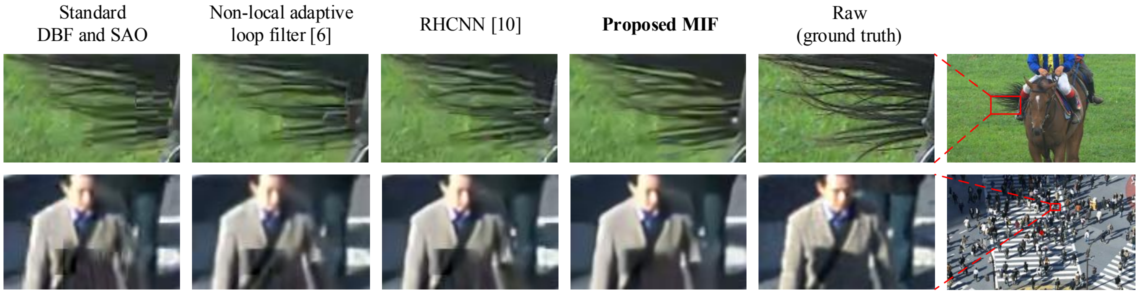

Figure 4: Comparison of subjective visual quality on sequences *RaceHorses* (Class C) and *PeopleOnStreet* (Class A) at QP = 37.

**Subjective visual quality.** Figure 4 illustrates the subjective visual quality among all four approaches. It can be observed that the frames enhanced by our approach remain less distortion than those by other approaches, e.g., the clearer edge of the horse tail and the reduced blocking artifacts on the pedestrians. The highest visual quality mainly benefits from the utilization of multiple adjacent frames in the proposed MIF approach.

## 4 Conclusion

In this paper, we have proposed a DenseNet based in-loop filter for HEVC. Different from existing in-loop filter approaches based on a single frame, our MIF approach enhances the quality of each encoded frame leveraging multiple adjacent frames. To this end, we first propose an RFS to find higher-quality frames. Then, we develop an MIF-Net model for multi-frame in-loop filter in HEVC, which is based on the DenseNet and benefits from the improved generalization capacity and computational efficiency. Finally, experimental results demonstrate that our approach achieves $-11.62\%$ of BD-BR saving and 0.39dB of BD-PSNR increment on average, outperforming the HM baseline and other state-of-the-art approaches.

## Acknowledgment

This work was supported by NSFC under Grants 61876013 and 61573037, and by the Fok Ying Tung Education Foundation under Grant 151061.

## References

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

[2] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, Dec 2012.

[3] C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, C. Tsai, C. Hsu, S. Lei, J. Park, and W. Han, "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1755–1764, Dec 2012.

[4] C. Tsai, C. Chen, T. Yamakage, I. S. Chong, Y. Huang, C. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz, and S. Lei, "Adaptive loop filtering for video coding,"

*IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 934–945, Dec 2013.

[5] M. Matsumura, Y. Bandoh, S Takamura, and H. Jozawa, "In-loop filter based on non-local means filter," JCTVC-E206, ITU-T SG16, Geneva, Swizerland, March 2011.

[6] X. Zhang, R. Xiong, W. Lin, J. Zhang, S. Wang, S. Ma, and W. Gao, "Low-rank-based nonlocal adaptive loop filter for high-efficiency video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2177–2188, Oct 2017.

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

[8] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for HEVC compressed videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[9] 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 *2018 Data Compression Conference*, March 2018, pp. 187–196.

[10] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, "Residual highway convolutional neural networks for in-loop filtering in HEVC," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3827–3841, Aug 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 770–778.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[13] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 4700–4708.

[15] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 2848–2857.

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[17] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 4724–4732.

[18] J. 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 Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec 2012.

[19] Xiph.org, "Xiph.org video test media," `https://media.xiph.org/video/derf`, 2017.

[20] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 3, pp. 475–489, June 2014.