Paper:

A Variable-Small-filter-size Residue-learning Dense CNN for Reconstruction of HEVC Video Frames

Huanhuan Chai*, Zhaohong Li, Zhenzhen Zhang***, and Shutong Xu*****

*,**,****Beijing Jiaotong University, Beijing, China
E-mail: zhhli2@bjtu.edu.cn
***Beijing Institute of Graphic Communication, Beijing, China
[Received 00/00/00; accepted 00/00/00]

Abstract. The in-loop filter module of High Efficiency Video Coding (HEVC) standard improves the reconstruction quality of compressed video frames, but it also brings bitrate increasing. In this paper, we aim to replace the existing in-loop filter module of HEVC with convolutional neural network (CNN) to facilitate HEVC intra coding performance on both visual quality and bitrate. First, two consecutive 3×3 convolutional layers are adopted instead of original 5×5 convolutional layer in Variable-filter-size Residue-learning CNN(VRCNN) to increase the nonlinearity and expression ability, then we incrementally utilize dense connection to boost unimpeded information flow among three blocks. Finally, the proposed reconstruction network called Variable-Small-filter-size Residue-learning Dense CNN (VSRDCNN) is obtained. The dataset is gotten by HEVC compression which turns off the in-loop filter module with four quantization parameters (*QPs*). In a progressive way, the weights of VSRDCNN on the highest *QP* 37 are obtained by training and are used as initial weights for VSRDCNN on the other three *QPs*. At last, the trained CNNs are used to replace the in-loop filter in HEVC to get better restoration performance. The proposed VSRDCNN is validated both subjectively and objectively via extensive experiments on HEVC standard test sequences. Experimental results show that the proposed VSRDCNN outperforms HEVC 0.30 dB on *BDpsnr* and 0.51% reduction on *BDrate*, and achieves the lowest *BDrate* compared with the state-of-the-art work.

Keywords: Convolutional Neural Network (CNN), High Efficiency Video Coding (HEVC), in-loop filter, super resolution

1. Introduction

The video without compression will occupy amount of size, resulting in taking more amount of disk space and timing. In recent years, the dramatical increase of video content presents great significance and pressure on video compression which enables to reduce the transmission bitstream while ensuring the visual quality. To address this, video coding standards have been initiated including Advanced Video Coding (AVC/H.264), High Efficiency Video Coding (HEVC/H.265) and Versatile Video Coding (VVC/H.266). These coding standards all employ a similar framework called as hybrid video coding as illustrated in Figure 1 and improve compression performance generation by generation. Hybrid coding is a combination of prediction coding and transform coding. Inter-frame prediction based on motion compensation is used to compress temporal redundancy, and transform such as DCT is used to compress spatial redundancy. The compressed original frame is restored to reconstructed frame by dequantization and inverse DCT.

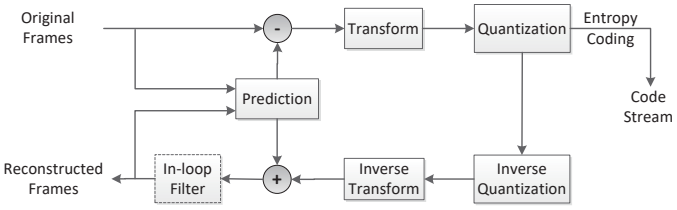

Fig. 1. : The workflow of hybrid video coding

However, in some senses, compression and ensuring visual quality are at the two ends of a spectrum of problems, i.e., there are distortion and artifacts especially at low bitrate in the process of compression including blocking artifact which is visible discontinuities at block boundaries owing to block-based coding, blurring which is as a result of losing high-frequency components, ringing, color bias and so on. These artifacts may greatly decrease the visual perception quality of reconstructed image or video frames. Therefore, how to reduce or eliminate these artifacts plays an important role in video frame reconstruction and has been extensively researched in the literature. In-loop filtering module of above coding standards is served as diminishing distortion and artifacts which mainly consists of two post-processing techniques namely deblocking filter (DBF) and sample adaptive offset (SAO) in both HEVC and VVC. DBF is designed to adaptively suppress artifacts along block boundaries using low-pass smoothing filters and not requires any additional bit. SAO which demands for additional bit is invoked after DBF to make a nonlinear adjustment that adds offsets to samples for in-

hibiting general compression artifacts. Adaptive loop filter (ALF) is the unique and the third technique in VVC based on HEVC, which minimizes distortions between the original and reconstructed blocks using an adaptively trained low-pass filter. All of these can suppress the artifacts for the major cases. However, the performances are still far from expectation, especially at low bit rates.

Convolutional neural network (CNN), as the most popular model of deep learning, has made great progress in the analysis, recognition and processing of images and videos. It has also been used in restoration of compressed image and video to depress distortion and artifacts, achieving improvement of visual perception quality, and saving of bitrate compared with above in-loop filter module in current video coding standards.

In the codec combined with CNN, there are mainly two cases in the restoration of compressed video frames. One is to replace some composition of in-loop filter module with designed CNN which can avoid bitrate increase because of omitting SAO of in-loop filter but it requires comprehensive network to restore video frame with good visual quality. Amongst networks which belong to this, Variable-filter-size Residue-learning CNN(VRCNN) [1] adopted variable filter size based on ARCNN [2] to achieve higher bitrate reduction, lower memory, and multiple computational speedups. MMS-net [3] which consisted of two sub-networks of different scales exploited the compression parameters from coding tree units (CTU) as input to alleviate blocking artifacts on the reconstructed images. The other is to post-process the video frame with designed CNN which can keep the reconstruction effect of original in-loop filter module but it brings the bitrate raising. Amongst networks which belong to this, RHCNN [4], following in-loop filter, was composed of several residual highway units and convolutional layers which allowed unimpeded information across several layers. DRCNN [5] which was between DBF and SAO took advantages of dense shortcuts and residual learning and exploited the multi-level features to fused the hierarchical feature. CACNN [6], after SAO, proposed content-aware multimodal filtering mechanism to realize the restoration of different regions which were the different Coding Tree Unit (CTU). MFRNet [7], for both post-processing and in-loop filter in video compression, contained four MFRBs which was connected using a cascading structure to reuse high dimensional features.

We found the previous VRCNN which adopted variable filter sizes was very fit to the variable block partitions of HEVC, so when it was used to substitute for the in-loop filter module, it achieved good visual performance and on average 4.6% bitrate reduction compared to HEVC baseline. Stimulated by this, an improved CNN architecture based on VRCNN is proposed in this paper. We first design an improved CNN named as Variable-Small-filter-size Residual-learning CNN(VSRCNN) which replaces 5×5 convolutional layer of VRCNN by two consecutive 3×3 convolutional layers for raising the nonlinearity of model. Then, a further improved CNN called as Variable-Small-filter-size Residue-learning Dense CNN (VSRD-

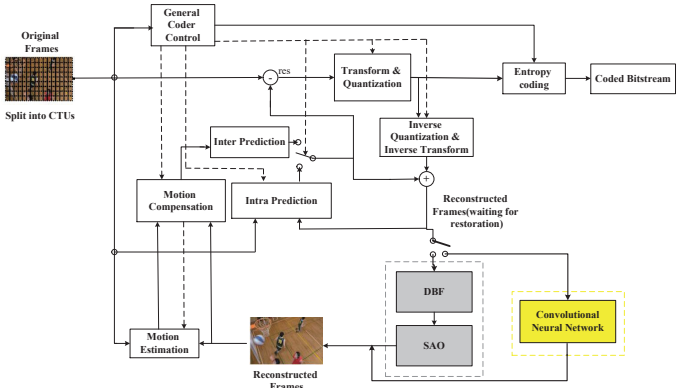

Fig. 2. : The pipeline of HEVC combined with CNN

CNN) employing residual and dense connection is proposed to increase the feature expression ability of the network. Finally, the proposed VSRDCNN has been integrated into HEVC intra coding for gradually improving the coding performance. Compared with the original in-loop filter module of HEVC, the proposed super resolution CNN not only realizes the bitrate saving under the same visual quality, but also realizes the visual quality improvement under the premise of the same bitrate. Furthermore, the proposed VSRDCNN demonstrates superior performance when it is compared to other restoration CNNs including VRCNN.

The remainder of the paper is organized as follows. Section 2 presents the details of proposed VSRCNN and VSRDCNN. Together with discussing the specifics of training and using designed CNNs, Section 3 reports the experimental results, followed by conclusion in Section 4.

2. Designed Networks

In this section, we will first point out the position of our network in HEVC. The following will be the demonstration of three networks of which the first is VRCNN proposed in literature [1], the remaining is our own designed network including the structure, configuration, and the reason why we do this.

2.1. CNN-based Coding Module

The CNN-based in-loop filter module used in HEVC is illustrated in Figure 2. We train our neural network in advance and then integrate it into HEVC whose DBF and SAO are turned off. The compressed frame waiting for restoration is fed into CNN (yellow box) rather than original in-loop filter module (gray box), producing the final reconstructed frame.

2.2. VRCNN

There are four layers in VRCNN which are considered as the four steps to reduce artifacts: feature extraction, feature enhancement, mapping, and reconstruction. The

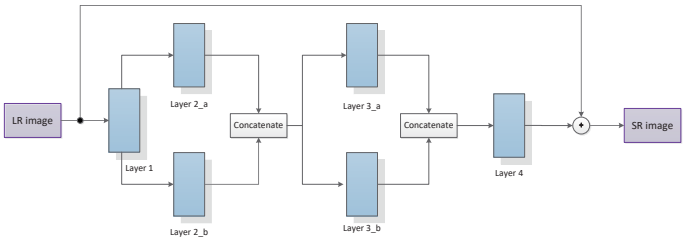

Fig. 3. : The structure of VRCNN

Table 1. : The configuration of VRCNN

Layer	1	2_a	2_b	3_a	3_b	4
Conv	1	2	3	4	5	6
Filter Size	5×5	5×5	3×3	3×3	1×1	3×3
Channels	64	32	16	48	16	64

structure of VRCNN is shown in Figure 3 and its configuration is given in Table 1.

One of the contributions of VRCNN is the diversity of filter size which is similar to the famous network called as GoogLeNet [8] that uses the combination of different size filters to extract different scale information of input image and achieves an excellent result on image classification. At the same time, the variable filter sizes correspond to the variable coding blocks in HEVC on which the transform is based and why the blocking effect amongst artifacts is produced. Quantization of the transformed coefficients causes distortion. Therefore, VRCNN reduces artifacts by using a combination of features extracted by two-size filters. The combination of 5×5 filter and 3×3 filter in the second layer plays an important role in making the "noisy" feature "cleaning" which is equivalent to denoise the feature maps. The third layer assembles 3×3 filter and 1×1 filter is designed to perform restoration of features. The reason why variable-filter-size method is not used in the first layer and the fourth layer is that these layers implement feature extraction and reconstruction respectively, which is not affected by the variable-size blocks in HEVC.

In addition, another technique of VRCNN is the skip connection between input and output which is the contribution of ResNet [9], enabling the whole network to learn the residue. Compared with the network learning an identity mapping between input and output, learning residue, i.e., the high frequency component of image is easier, because it is difficult for the former to not appear degradation in the process of network training. However, by using the latter, residual learning ensures the network signal can back-propagate to previous layer directly.

2.3. The Improved VSRCNN

We redesigned the convolutional kernel size of VRCNN named as Variable-Small-filter-size Residue-learning CNN (VSRCNN) to increase nonlinearity and

Table 2. : The configuration of VSRCNN

Layer	1	2_a	2_b	3_a	3_b	4
Conv	1/2	3/4	5	6	7	8
Filter Size	3×3	3×3	3×3	3×3	1×1	3×3
Channels	64/64	64/32	16	48	16	64

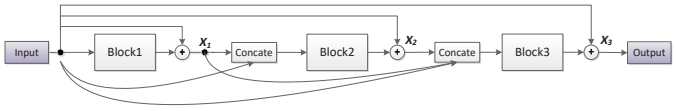

Fig. 4. : The structure of VSRDCNN

expression ability of the network whose base structure shares with that of VRCNN in Figure 3. What is mainly different from VRCNN in the configuration is given in Table 2.

As shown in Figure 3 and Table 2, we replace the entire 5×5 conv module of VRCNN with two consecutive 3×3 conv modules in VSRCNN. Note that the conv module includes a convolution layer followed by batch normalization and activation function which is the ReLU. Especially, we put the ReLU activation function of eighth conv module behind the skip connection. Meanwhile, it is attractive in practice that we keep shortcut of VRCNN to prevent gradient vanishing which introduces neither extra parameter nor computation complexity.

How VSRCNN improves performance of network is described in detail as follows. Convolutional kernel as the foundation of convolutional layer can extract different kinds of feature with various weights. By changing the size of kernel, we can extract local features of different size. i.e., the feature learned by 5×5 filter is more global than that through 3×3 filter, however, by adding another 3×3 filter after the first 3×3 filter we can learn from the front layer the same feature as that of 5×5 filter. What a single 5×5 filter learns is equivalent to two stacked 3×3 filters. The advantage of two consecutive 3×3 convolutional layers is it can enable network to use the activation function one more time and learn more kinds of feature which increase the nonlinearity and expression ability.

2.4. The Further Improved VSRDCNN

Based on VSRCNN, we add the depth of the network and redesign a new network called as Variable-Small-filter-size Residue-learning Dense CNN (VSRDCNN).The structure of VSRDCNN contains three blocks which is exhibited in Figure 4. Each block is composed of the four layers of VSRCNN. The blocks are linked by dense connection and a jump connection is added between the output of each block and the original input.

This paragraph elaborates how VSRDCNN uses residual connection and its advantage. The output of each block is added a jump connection with the original input so that the block learns the residue with a weak differ-

ence between the output of current block and original input rather than directly learn the identity mapping, which caters to the problem that the network is difficult to train with the increase of depth. It is noteworthy that the current unit combines the features learned by all previous blocks at the same time. As for adding jump connection, it is easy to intend the output of current block to be closer to the original input rather than the input of the current block.

Next, how VSRDCNN uses dense connection [10]and its advantages are given. The input of each block is the concatenation of the output of all previous blocks together with the original input. The feature extracted from each layer is equivalent to a nonlinear transform of the input data. With the increase of CNN depth, it gradually fits more nonlinear functions and the complexity of the transform dramatically grows. Dense connection, further improves the information flow among blocks ensures the current block can comprehensively utilize the features extracted from previous blocks with low complexity. In other words, we combine the shallow features with the deep features in the current block, which makes it easy to get a smooth decision function with better generalization performance. Compared with learning the shallow features and deep features of input separately and then hanging them together, our dense connection effectively reduces the amount of network parameters.

3. Results and Discussion

In this section, we test our proposed CNNs with some competing methods over HEVC reference software HM16.15, downloaded from HEVC official website. We first outline our experimental setup including dataset and loss function used in training and validation. Then we give the using of CNNs in testing. Next, for verifying the superiority and robustness of our CNNs, we provide the results generated by all methods on both subjective and objective criteria. Finally, the comparison of our CNNs with other CNNs in literature is given.

3.1. Configurations

For dataset, we make pairs of images as training set and validation set according to DIV2K [11] dataset. An original image $\boldsymbol{HR}_n$ (High Resolution), where $n \in \{1,...,900\}$ indexes each image, is compressed with HEVC intra coding while turning off the in-loop filter. The compressed image called as $\boldsymbol{LR}_n$ (Low Resolution) is regarded as the input of the network and $\boldsymbol{SR}_n$ (Super Resolu-tion) as the output of CNN. As a result of the limitation of GPU, we crop $\boldsymbol{HR}$ and $\boldsymbol{LR}$ to 128×128 sub-images without overlapping. In total, we get 95,115 pairs of training samples which will be randomly shuffled and the mini-batch is 64 when training and 11,910 pairs of validation samples.

For loss function, we minimize the following Mean Square Error (MSE) as the objective of training:

$$L(\Phi) = \frac{1}{N}\sum_{n=1}^{N} \|F(LR|\Phi) - HR\|_2^2 \quad . \ . \ . \ . \ . \quad (1)$$

Where $\boldsymbol{F(LR|\Phi)}$ is equivalent to $\boldsymbol{SR}$ (Super Resolution) which is output of the trained CNN, Φ consisting of $\boldsymbol{W}_i$,$\boldsymbol{B}_i$, with $i \in \{1,2,3,4\}$ indicating which layer it represents is the whole parameter set of CNN, N is the total pixels of $\boldsymbol{SR}_n$ or $\boldsymbol{HR}_n$ image. The loss is minimized using adaptive moment estimation (Adam) [12] as the stochastic gradient descent optimization algorithm [13].

We use the deep learning library Pytorch on a GeForce GTX 1080 Ti with CUDA version 11.0 to implement all operations in our networks. We train four times for each network at *QPs* 22, 27, 32 and 37. The momentum is set to 0.9 and weight decay is 10^{-4}. A network model with specific parameters at *QP* 37 is done through 160 epochs setting the learning rate of 10^{-4}. To accelerate the converge speed, we fine-tune the model at *QPs* 22, 27, 32 which is initialized with parameters of trained model at *QP* 37. This is empirically better than train from scratch.

Note that the trained model of four *QPs* for our VSRCNN and VSRDCNN is based only on the *Y* luminance channel (i.e., the *Y* out of *YUV*). However, we test it not only on *Y* channel but also on chrominance channels (*U* and *V*) using the HEVC standard test sequence to prove the generalization ability of network. We integrate the trained model into HEVC reference software HM16.15 by libtorch and OpenCV for intra coding with the original DBF and SAO turned off and test it on the HEVC standard test sequence which contains five classes,18 sequences shown in Table 3 and only the first frame is used as evaluation dataset.

3.2. Subjective Quality Comparison

We compare the visual quality of reconstructed images as shown in Figure 5 and show the same 64×64 block in the first frame of BlowingBubbles. It can be observed that the image compressed by HEVC with DBF and SAO turned off has obvious blocking and ring artifacts. Through in-loop filter reconstruction of HEVC coding, the restoration frame of BlowingBubbles at *QP* 37 greatly reduces the blocking, but ringing is still visible. The processed frames by VRCNN suppress all kinds of artifacts and produce a good visual quality. Our designed VSRCNN is better than VRCNN and VSRDCNN is further better than VSRCNN especially from the left eyebrow in the dotted box of the girl in Figure 5.

3.3. Objective Quality with HEVC Baseline

In order to objectively measure the performance of the proposed CNN, three common evaluation criteria are chosen: *PSNR*, *BDpsnr*, *BDrate*.

3.3.1. Objective Evaluation on *PSNR*

The first index *PSNR* is a classic estimate for objective quality between $\boldsymbol{SR}_n$ and $\boldsymbol{HR}_n$ images. For an 8-bit depth image, the formula of *PSNR* is given as follows:

$$MSE = (\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(SR(x,y) - HR(x,y))^2)/(W \times H)$$
$$PSNR = 10 \times log_{10}(255^2/MSE) \quad (2)$$

Table 3. : The configuration of the evaluation dataset

Class	Resolution	Sequence
Class A	2560×1600	PeopleOnStreet, Traffic
Class B	1920×1080	BasketballDrive, BQTerrace, Cactus, Kimon, Park Scene
Class C	832×480	BasketballDrill, BQMall, PartyScene, RaceHorses
Class D	416×240	BasketballPass, BlowingBubbles, BQSquare, RaceHorses
Class E	1280×720	FourPeople, Johnny, KristenAndSara

(a) ***HR*** (b) ***LR*** (c) HEVC (d) VRCNN (e) VSRCNN (f) VSRDCNN

Fig. 5. : One block in the first frame of BlowingBubbles compressed at *QP*=37 and reconstructed by HEVC baseline as well as three CNNs

$$PSNR_{YUV} = (6 \times PSNR_Y + PSNR_U + PSNR_V)/8 \quad (3)$$

Where ***W*** and ***H*** are on behalf of the width and height of the image, respectively. With the *YUV* format of our test set which comprises of three components, i.e., *Y*, *U* and *V*, as the HM16.15 does, we use a weighted sum of the average *PSNR* values for luma and chroma components, which is defined by equation (3) as follows. Where the $PSNR_Y$, $PSNR_U$ and $PSNR_V$ are calculated by equation (2) using the individual color component *Y*, *U* and *V*. In a way, the *PSNR* reflects the pixel-wise similarity between ***HR*** and ***SR***. Note that we train CNNs on *Y* channel and test it on *YUV* channels to prove the generalization ability of the networks. The results of HEVC, HEVC_off which turns off the DBF and SAO, VRCNN and the proposed two networks at *QP* 37 are recorded in Table 4. Note that the performance of network at the other three *QPs* is similar to that at *QP* 37. The *PSNR* of reconstructed images using VRCNN, VSRCNN, and VSRDCNN are all higher than that of reconstructed images using in-loop filter from either the average value of all classes or the average value of each class. From the view of the average *PSNR* of all classes on the *YUV* channels in Table 4, the good generalization performance of proposed CNNs can be seen. The value of HEVC is 0.27dB higher than that of HEVC_off which embodies the effectiveness of DBF and SAO module, the value of VRCNN is 0.04dB lower than our VSRCNN owing to its two stacked 3×3 convolutional layers, VSRCNN's value is 0.09dB lower than that of VSRDCNN which benefits from dense connection among blocks.

In addition, we add the experiment to compare the reconstruction effect of the proposed VSRDCNN and the DBF and SAO of HEVC on different video content. First, 18 frames ***LR*** is gotten from encoding above 18 frames ***HR*** evaluation da-taset by HEVC at *QP* 37 which turns off the DBF and SAO. Then, we crop the ***LR*** frames to 128×128 blocks and obtain 1314 blocks in total. Next, we use the edge extraction operator to sum the edge pixel numbers of each block to get a score, and then sort all blocks into three groups according to the scores from high to low. The three groups are high-texture blocks, middle-texture blocks, and low-texture blocks in which the high-texture blocks have the greatest number of edge pixels indicating complicated texture areas, and the low-texture blocks have the least number of edge pixels indicating flat areas. The mean $\Delta PSNR$ value which equals to the *PSNR* of reconstructed blocks by the proposed VSRDCNN minus the *PSNR* of reconstructed blocks by default HEVC coding for the three groups is calculated and shown in Table 5. From the result, we can observe that it has the best restoration effect of 0.33 dB increasing on the middle-texture blocks. The reason why it has the least effect of 0.18 dB on the high-texture blocks is that the complicated texture is lost seriously in the HEVC encoding, so it is hard for reconstruction. As for the low-texture blocks, the result is 0.11 dB less than that of high-texture blocks, because it has few textures in flat areas, so the reconstruction effect is not obvious. Based on the conclusion, adopting different CNNs for different content videos or blocks may get more comprehensive restoration results.

3.3.2. Objective Evaluation on *BDpsnr* and *BDrate*

For a compressed video, what shows the new method has better performance is that the *Bit Rate* decreases as well as the *PSNR* increases. However, there will be a situation where the *Bit Rate* is lower than the original method, but the *PSNR* reduces. In this case, *BDrate* and *BDpsnr* based on four *QPs* which mentioned in [14] are needed to measure the quality of the method.

BDpsnr which is used as the second index is the difference of *PSNR* values of compressed video between cur-

Table 4. : The *PSNR* of different CNNs on *YUV* channels

Class	*PSNR*_yuv (dB)				
	HEVC	HEVC_off	VRCNN	VSRCNN	VSRDCNN
Class A	35.097	34.748	35.378	<u>35.422</u>	**35.510**
Class B	35.368	35.119	35.484	<u>35.501</u>	**35.618**
Class C	32.301	32.097	32.642	<u>32.683</u>	**32.730**
Class D	31.922	31.735	32.383	<u>32.436</u>	**32.472**
Class E	37.566	37.236	37.873	<u>37.920</u>	**38.089**
Average	34.45	34.18	34.75	<u>34.79</u>	**34.88**

Table 5. : The mean value of Δ*PSNR* on three groups of different video content

	high-tex block	mid-tex block	low-tex block
Δ*PSNR*	0.18	**0.33**	0.22

rent method and HEVC baseline under the condition of the same *Bit Rate*. The larger the *BDpsnr* is, the smaller the quality loss of the current method is. The *BDpsnr* results of three CNN-combined HEVC coding methods are shown in Table 6 from which the VSRDCNN's is 0.02dB higher than VSRCNN's on average of five classes, and VSRCNN's is 0.06dB higher than VRCNN's under the same condition on *YUV* channels. Those good results profit from the stacked two small filters of VSRCNN increasing feature representation and nonlinearity of network and the dense connection of VSRDCNN promoting the unimpeded transmission of information while deepening the network.

BDrate which is the third criterion in this paper is the percentage of *Bit Rate* saving by current coding method under the same *PSNR*. *BDrate* is usually negative. The smaller absolute value of *BDrate* is, the better compression performance of the current encoder is. The *BDrate* results are recorded in Table 6 from which we can observe that the largest absolute value is VSRDCNN's in bold of -4.59%, the second largest is VSRCNN's underlined of -4.22% on the *YUV* channels , meanwhile, VRCNN's mean value of Class B on *YUV* channels even is positive. Although we train the model on the *Y* channel, it can be generalized to *YUV* channels. In summary, the proposed VSRCNN and VSRDCNN enable to achieve *Bit Rate* saving step by step thanks to the ingenious design of CNN.

3.3.3. Computational Complexity Analysis

In this section, the computational complexity which affects the codec's real-time performance will be analyzed.Table 7 shows the average time per frame for each class video. As can be seen from the table, a video with higher resolution needs more time to compress. The encoding time used by different CNNs is all higher than that of default HEVC. Note that it requires connection process of CNN(trained based on Pytorch) and HEVC codec(HM16.15 completed by c++). In addition to the prediction of CNN model which takes about 15% of the entire time, the other 85% is spent in preprocess and postprocess during connection. Therefore, the time costs can be reduced by further optimizing the connection. When comparing among different CNNs, VSRCNN takes 0.21 seconds less than VRCNN which benefits from the stacked two small filters of VSRCNN. VSRDCNN based on VSRCNN which contains three blocks deepens the network depth at the expense of increasing the encoding time by 0.3 seconds on average compared with VRCNN. On the whole, the encoding time taken by three CNNs is similar. However, VSRDCNN is the best and VSRCNN is the second-best from the view of other performances such as improving visual quality and reducing bitrate.

3.4. Objective Evaluation Compared with Other Networks

As mentioned in above context, some restoration CNNs have been developed to replace in-loop filter for achieving better coding efficiency, in which RHCNN [4], DRCNN [5], CACNN [6] and VRCNN [1] are the latest and superior networks, so the proposed two CNNs are compared with them in terms of *BDrate*. Due to the limitation of results available in the literature, only results of Class C and Class D are compared in Table 8. The bold number is the best and the underlined is the second-best in the current line. The *BDrate* of each method is negative, which indicates that all designed restoration CNNs have better performance than the in-loop filter module of HEVC. The performance of the proposed VSRDCNN is superior to the proposed VSRCNN of 0.3% *BDrate* saving and the VSRCNN exceeds all the other methods even can achieve 2.05% *BDrate* reduction compared with CACNN [6]on the average of the two Classes in Table 8. It shows that replacing the 5×5 convolutional layers with two consecutive 3×3 convolutional layers of VSRCNN is beneficial, and further, the stacked three blocks with VSRCNN through dense connection of VSRDCNN is also successful.

4. Conclusion

In this paper, we present two progressive networks, namely as VSRCNN and VSRDCNN, to replace the DBF

Table 6. : The *BDpsnr* and *BDrate* of different CNNs on *YUV* channels

Class	*BDpsnr*_yuv(dB)			*BDrate*_yuv (%)		
	VRCNN	VSRCNN	VSRDCNN	VRCNN	VSRCNN	VSRDCNN
Class A	0.173	0.212	**0.229**	-3.469	-4.214	**-4.550**
Class B	0.003	0.039	**0.055**	0.156	-0.920	**-1.421**
Class C	0.278	0.331	**0.349**	-4.807	-5.736	**-6.074**
Class D	0.364	0.420	**0.436**	-5.480	-6.281	**-6.512**
Class E	0.070	0.190	**0.201**	-1.359	-3.959	**-4.406**
Average	0.17	0.23	**0.25**	-2.99	-4.22	**-4.59**

Table 7. : The encoding time(s) per frame on HEVC and different CNNs with *QP*=37

Class	Resolution	HEVC	VRCNN	VSRCNN	VSRDCNN
Class A	2560×1600	19.43	38.06	37.18	38.68
Class B	1920×1080	9.55	18.45	18.42	18.82
Class C	832×480	2.25	4.4	4.34	4.96
Class D	416×240	0.53	1.53	1.46	1.46
Class E	1280×720	4.25	8.45	8.41	8.45
Average		7.20	14.17	13.96	14.47

Table 8. : The comparison with other CNNs on *BDrate*

	RHCNN [4]	DRCNN [5]	CACNN [6]	VRCNN [1]	VSRCNN	VSRDCNN
Class C	**-7.10%**	-3.90%	-4.50%	-4.80%	-5.70%	-6.00%
Class D	-4.40%	-4.60%	-3.30%	-5.40%	-6.20%	**-6.50%**
Average	-5.75%	-4.25%	-3.90%	-5.10%	-5.95%	**-6.25%**

and SAO module of HEVC for enhancing intra coding performance. The proposed VSRCNN replaces original filter in VRCNN with two small filters to improve feature expression ability. Further, VSRDCNN which takes VSRCNN as basic unit comprehensively utilizes features and boosts unimpeded information flow through dense connection. By testing the proposed CNNs on HEVC standard video sequences, effectiveness of our network is proved via extensive experiments on both quantitative and qualitative evaluation. Compared with HEVC baseline and other latest reconstruction CNNs, the proposed VSRDCNN achieves the best performance on both visual quality and bitrate saving. Our future work is to design different CNNs for different image blocks according to their texture, as we found that the proposed VSRDCNN has different effects for image blocks with high-texture, middle-texture, and low-texture.

Acknowledgements

This work was supported by The Scientific Research Common Program of Beijing Municipal Commission of Education (No. KM202110015004)

References:

[1] Dai Y, Dong L, Feng W. A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding [J]. International Conference on Multimedia Modeling, 2017.

[2] Dong C, Deng Y, Loy C C, et al. Compression Artifacts Reduction by a Deep Convolutional Network [J]. IEEE, 2015.

[3] Kang J, Kim S, Lee K M . Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec[C]// 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2018.

[4] Zhang Y, Shen T, Ji X, et al. Residual Highway Convolutional Neural Networks for in-loop Filtering in HEVC[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2018:3827.

[5] Wang Y, Zhu H, Li Y, et al. Dense Residual Convolutional Neural Network based In-loop Filter for HEVC[C]// 2018 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2019.K. Biswas.

[6] Jia C, Wang S, Zhang X, et al. Content-Aware Convolutional Neural Network for In-loop Filtering in High Efficiency Video Coding[J]. IEEE Transactions on Image Processing, 2019:1-1.

[7] Ma D, Zhang F, Bull D R. MFRNet: A New CNN Architecture for Post-Processing and In-loop Filtering[J]. 2020.

[8] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[9] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[10] Huang G, Liu Z, Laurens V, et al. Densely Connected Convolutional Networks[J]. IEEE Computer Society, 2016.

[11] Agustsson E, Timofte R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2017.

[12] Kingma D, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.

[13] Sutskever I, Martens J, Dahl G, et al. On the importance of initialization and momentum in deep learning[C]//International conference on machine learning. PMLR, 2013: 1139-1147.

[14] Bjontegaard G. Calculation of average PSNR differences between RD-curves[J]. ITU-T VCEG-M33, April, 2001, 2001.