IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON MOBILE MULTIMEDIA: METHODOLOGY AND APPLICATIONS

Received December 3, 2019, accepted December 19, 2019, date of publication December 23, 2019, date of current version January 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961760

A CNN-Based Post-Processing Algorithm for Video Coding Efficiency Improvement

HAIWU ZHAO[1], MING HE[1], GUOWEI TENG[1], XIWU SHANG[2], GUOZHONG WANG[2], AND YIYAN FENG[1]

[1]School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
[2]School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Corresponding author: Ming He (heming_199510@163.com)

ABSTRACT Lossy compression algorithms are widely used in video coding. However, lossy compressed videos exist some annoying distortion and artifacts, such as blocking, blurring, and ringing. Thus, coding efficiency improvement is a steady-state topic in the domain of video coding. High Efficiency Video Coding (HEVC), a recent video standard, adopts two in-loop filters for the improvement of the coding efficiency, including deblocking (DB) and sample adaptive offset (SAO). In a certain extent, traditional in-loop filters reduce the distortion and improve the video quality. But the reduction of the distortion is a nonlinear problem that is difficult to be solved by traditional linear filters. Recently, the progress of deep learning shows the possibility to settle the complex problems in the computer vision field. Meanwhile, according to the compressive sensing theory, the post-processing method at the decoder end can further enhance the coding efficiency. In this paper, we propose a variable-filter-size Residue-learning convolutional neural network with batch normalization layer (VRCNN-BN). Our model is an end-to-end model. We feed the decoded pictures to the model at the decoder end. Different from previous methods, we apply the model to luma pictures and chroma pictures, respectively. In order to comprehensively evaluate the coding performance of both luma and chroma components, the color-sensitivity-based combined PSNR (CS-PSNR) is exploited to measure the effectiveness of the proposed method. Compared to HEVC baseline, our approach achieves an average BD-rate reduction of 10.3%, 8.9%, 13.1% and 11.8% in terms of CS-PSNR for random access, all intra, low delay P and low delay B configurations, respectively. Abundant experimental results indicate that our method is better than existing similar methods.

INDEX TERMS Convolutional neural network, end-to-end, post-processing, high efficiency video coding.

I. INTRODUCTION

In the past decades, videos have taken an important part in our daily communications. Under the condition of bandwidth-limited transmission, video compression has become a significant research orientation of the video coding. Lossy video compression standards, targeting for saving coding bit-rates and transmitting high quality videos, have achieved a commendable compression efficiency, such as high efficiency video coding (HEVC) standard [1] and IEEE1857 video coding standard [2]. However, most of the lossy video coding algorithms are based on block-prediction, block-transform coding, and quantization coding. Thus, the lossy video coding algorithms by nature cause distortion and artifacts. In the current video coding standard, some post-processing algorithms are committed to reducing distortion and artifacts, such as sample adaptive offset (SAO) [3] and deblocking (DB) [4]. Meanwhile, in pace of the development of deep learning, it becomes feasible to apply convolutional neural networks (CNNs) to end-to-end post-processing methods for videos and images [5]–[7].

In this paper, we propose a post-processing method for luma pictures and chroma pictures. In contrast to earlier CNN-based post-processing algorithms, our method takes luma pictures and chroma pictures into the post-processing model, respectively. In order to satisfy the requirements, we propose a modified variable-filter-size residue-learning convolutional neural network (VRCNN). Our proposed model adds the batch normalization layer [8] to VRCNN, named VRCNN-BN. The structure of VRCNN-

The associate editor coordinating the review of this manuscript and approving it for publication was Honggang Wang.

BN will be described in detail in section III. In our model, we implement an end-to-end post-processing method for video coding. Besides, VRCNN-BN model is not sensible to the size of input pictures. The model can apply to the videos, which are in the usual case of 4:2:0 color sampling [1]. In the test, each sequence is compressed by HEVC under four coding configurations: all intra (AI), low delay P (LDP), low delay B (LDB), and random access (RA) [9]. The compressed videos are then divided into luma pictures and chroma pictures, which are post-processed respectively. In the experiment, the chroma pictures tend to perform better than the luma pictures. The results of the experiment verify that our model can apply to video post-processing problems comprehensively.

In the domain of post-processing for videos and images, most studies focus on reducing blocking artifacts, ringing artifacts, color biases, and blurring artifacts [10]. In HEVC, the most current video standard, there are mainly two post-processing algorithms for artifacts reduction, including DB and SAO. DB is specifically designed for reducing blocking artifacts, which lessens the difference at the boundaries of prediction blocks and transform blocks. Furthermore, DB does not require any additional bits [4]. SAO is applied after the DB to further reduce the artifacts by changing the sample offset of the blocks in an image. Compared to DB, SAO is designed to attenuate general compression artifacts. However, SAO requires to transmit additional bits, that are used in explicitly signaling the offset of each block from the encoder to the decoder for reducing sample distortion effectively [3]. Both DB and SAO contribute to improve video quality and save bit-rates efficiently. However, it is a complicated problem to reduce the distortion in lossy video compression. Both DB and SAO, the existing in-loop filtering methods, are not enough to satisfy the higher quality of video coding.

The main limitation of current post-processing algorithms in HEVC is that the linear filters cannot cope with nonlinear distortion. Traditional post-processing algorithms mainly focus on DB filters [11], [12]. In contrast to traditional methods, deep learning has led to a series of breakthroughs for dealing with nonlinear problems [7]. And most progress of deep learning is not only the result of more powerful hardware, lager dataset and deeper models, but also a consequence of new ideas and algorithms [5]. With the introduction of some new methods, such as batch normalization (BN) layer [8], activation function of rectified linear units (ReLU) [13], and optimizer of Adam [14], deep learning has dramatically advanced the state of the art in vision, speech, and many other areas[8]. As so far, many proposed CNNs perform better than current post-processing algorithms in HEVC. In terms of post-processing algorithms for videos, CNN-based approaches can roughly be divided into two aspects as follows:

1) the approaches are designed to replace in-loop filters at the encoder end [15]–[17]. Among current video standards, the in-loop filters are applied to the reconstructed samples before writing them into the decoded picture buffer in encoder loop [1]. And the picture buffer is used for reference buffer in prediction coding. Thus, the in-loop filters are encoder-end post-processing algorithms that improve the coding efficiency and change the bit-streams signaled to the decoder end.

2) the approaches improve video coding efficiency at the decoder end [18]–[21]. for some applications limited by the bandwidth and storage, the post-processing algorithms use the decoded videos as inputs. Although the in-loop filters reduce the distortion inside videos at the encoder end. However, it is hard to assure the optimal coding efficiency. Thus, the post-processing methods at the decoder end are possible to further enhance the quality of decoded videos.

In this paper, we mainly focus on the coding efficiency improved by CNN-based post-processing algorithms at the decoder end. In section II, inspired by related works, we propose a CNN-based post-processing method and a network structure, named VRCNN-BN. Section III presents the details of VRCNN-BN and the post-processing for luma pictures and chroma pictures. Experimental results are demonstrated in section IV. And section V concludes our work and presents the future works.

II. RELATED WORKS

Currently, with the development of high-performance GPU device, there are several existing CNNs for artifacts reduction and quality improvement. Some recent proposals and methods are summarized as follows, which have made some progress.

A. CNN-BASED POST-PROCESSING ALGORITHMS AT THE ENCODER END

In the domain of post-processing for videos, there are two aspects as mentioned previously. One is the method of in-loop filters for videos, which performs at the encoder end. Another is about out-of-loop post-processing algorithms that works at the decoder end. According to the research in residual learning, K. He et al. proposed a deep residual network for solving image recognition problems [7]. In this article, the ideas of residual learning influence the designing of models for CNN-based post-processing.

Among the in-loop filters, W. Park et al. proposed an in-loop filtering technique using convolutional neural network (IFCNN) for replacing SAO [15]. They show the results of reducing 2.8% in average BD-rate. J. Kang et al. proposed a multi-modal/multi-scale convolutional neural network (MMS-net) to replace exiting DB and SAO in HEVC [16]. Their method consists of two sub-networks of different scales, which reduces the average BD-rate by 4.55% and 8.5%, respectively. Y. Dai et al. proposed a Variable-filter-size Residue learning convolutional neural network (VRCNN) as the replacement of both DB and SAO in HEVC intra coding [17]. In their experiments, VRCNN

is reported to achieve a promising result of average 4.6% BD-rate reduction.

B. CNN-BASED POST-PROCESSING ALGORITHMS AT THE DECODER END

At the decoder end, the CNN-Based post-processing methods performer as out-of-loop filters. Compared to the in-loop filters at the encoder end, the out-of-loop filters, at the decoder end, apply to the decoded videos and satisfy the demands to improve the quality of decoded videos.

Since the video is encoded at the encoder end, the out-of-loop filters act on the decoded video, which are similar to the super resolution (SR) algorithms for videos or images. In terms of deep learning for SR problems, C. Dong et al. proposed a structure named super resolution convolution neural network (SRCNN) [22]. Based on SRCNN, C. Dong et al. proposed an artifacts reduction convolution neural network (ARCNN) for reducing artifacts in lossy JPEG [23] images [24]. In their article, the ARCNN is reported to achieve 1 dB higher than JPEG. Furtherly, J. Kim et al. proposed a very deep CNNs for SR problems (VDSR) inspired by VGG-net [6], which claims that VDSR performs better than SRCNN with extremely high learning rates (10^4 times higher than SRCNN) [25]. Based on the VDSR for SR problems, C. Li et al. proposed a VDSR, with 20 convolution layers [19], for post-processing algorithms. their approach is adopted to extract more meaningful information from the reconstructed error and improve the filtering performance. Compared to HEVC baseline, their approach achieves an average BD-rate reduction of 1.6% on the six sequences in 2017 ICIP Grand Challenge[26].

More recently, T. Wang et al. proposed a Deep CNN-based Auto-Decoder (DCAD) for post-processing at the decoder end [18]. At the decoder end, their approach can further improve the coding efficiency post the DB and SAO. In this paper, the DCAD is claimed to have an average BD-rate reduction of 5.0%, 6.4%, 5.3% and 5.5% for AI, LDP, LDB, and RA configurations, respectively. L. Ma et al. proposed a Residual-based Video Restoration Network (Residual-VRN) for video post-processing [20][21]. Compared to HEVC baseline, their method performs better than DCAD with an average BD-rate reduction of 7.4%,9.4%,7.4% and 7.6% for AI, LDP, LDB, and RA configurations, respectively.

Meanwhile, there are some of the latest methods for post-processing for images post-processing, which are similar to the post-processing algorithms for intra video coding. M. Barni et al. proposed a JPEG -aware CNN for JPEG post-processing, which is robust to JPEG compression [27]. H. Ma et al. proposed a pixel CNN for JPEG images post-processing [28]. In their article, the pixel CNN is reported to achieve 0.74 dB higher than JPEG-2000, averagely [29].

III. VRCNN-BN BASED POST-PROCESSING

In this section, we analysis the structure of VRCNN-BN. And then we propose our post-processing method that works on luma pictures and chroma pictures.

A. STRUCTURE OF VRCNN-BN

As mentioned in section I, distortion reduction is a complex nonlinear problem. Thus, in VRCNN-BN, we introduce the nonlinear function ReLU as the activation function. The ReLU works as the activation function to ensure that gradients always exist. As described in (1), the output feature maps of ReLU layer are always greater than or equal to zero. Since most input feature maps are composed of positive features, ReLU layer is a good guarantee for the transfer of gradients during back propagation. However, in VRCNN-BN, we train the residual information between low quality images with high quality images. In order to avoid dead ReLU layers, we only use the ReLU layer in the first three layers. As shown in Fig. 1, we keep the output feature maps of layer 4 without using the activation function.

$$ReLU\ (x) = \begin{cases} 0 & if\ x < 0 \\ x & if\ x \geq 0 \end{cases} \tag{1}$$

Meanwhile, we choose the BN layer as the normalization item in our model. As mentioned in [8], BN layer performs better than dropout and L2-norm since the BN layer learns the scale and shift on a batch size.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2}$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2 \tag{3}$$

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2}} \tag{4}$$

$$y = \gamma\hat{x} + \beta \tag{5}$$

As shown in (2), (3), (4), (5), the input x of BN layer is a N-dimensional input. And N is the size of a mini-batch. In the process of training VRCNN-BN, we set the size of a mini-batch is 120. we count the mean μ and the variance σ^2 of the mini-batch, respectively. We normalize each dimension of x and get the result $\hat{x}$ since the mean and the variance of the mini-batch have been counted. The $\hat{x}$ is also a N-dimensional variable. Then the BN layer introduces a pair of parameters γ and β. γ and β scale and shift the normalized value $\hat{x}$. Finally, we get the y which is a N-dimensional output of BN layer. The above is the entire process of the BN layer. During the actual use of BN layer, a feature map shares a set of BN layer parameters. And BN layer is not sensitive to the size of input feature maps. Thus, we can resize the input feature maps in the case of using a set of BN layer parameters. In this way, we can use the trained model to process videos of different resolutions.

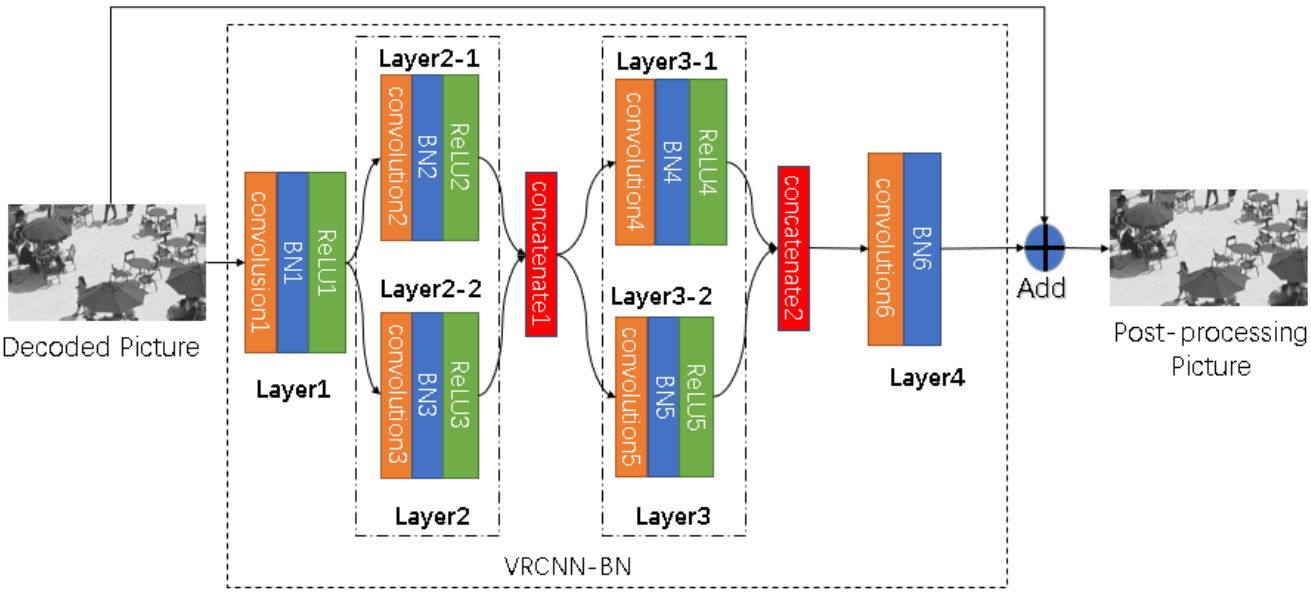


FIGURE 1. The structure of VRCNN-BN: there are four layers in VRCNN-BN, including six combination layers (each combination layer consists of a convolution layer, a BN layer and a ReLU layer except layer 4). Between layer 2 and layer 3 and between layer 3 and layer 4, there is a concatenate layer, respectively, which combines the output feature maps of pervious layer. At the end of VRCNN-BN, we add the output feature map with decoded picture, and the result is post-processing picture.

TABLE 1. The structure of vrcnn-BN and some Data for each layer (including filter size, number of filters, number of parameters, AND which layer connected to).

Layers		filter size	filters	parameters	Connected to
Input		-	-	-	-
Layer1		3x3	64	896	Input
Layer2	Layer2-1	5x5	16	25680	Layer1
	Layer2-2	3x3	32	18592	Layer1
Concatenate1		-	-	-	Layer2
Layer3	Layer3-1	3x3	16	6992	Concatenate1
	Layer3-2	1x1	32	1696	Concatenate1
Concatenate2		-	-	-	Layer3
Layer4		1x1	1	53	Concatenate2
Add		-	-	-	Layer4, Input
output		-	-	-	Add
Total		-	-	53909	-

The structure of VRCNN-BN is shown in Fig 1 and Table 1. As shown in Fig 1, VRCNN-BN extract the residual image between the input image and the output image. And in layer 2 and layer 3, there are two filters of different sizes in each layer. Then the concatenate layer combines the output feature maps of two filters at the channel dimension. For example, when layer 2-1 and layer 2-2 have 16 and 32 output feature maps respectively, concatenate1 has 48 output feature maps. The number of output feature maps in concatenate 2 is 48 in the same reason. In Table 1, the parameters of each layer are also shown. By training these parameters, the output images converge to the label images. After the training process, the model can be used to post-process the low-quality images. The process of training and testing will be introduced in detail in section IV.

B. PROPOSAL POST-PROCESSING METHODS

As mentioned in section II, previous post-processing methods mainly focus on the post-processing of luma pictures.

As shown in Fig 2, our approach provides the VRCNN-BN-based schemes for luma pictures and chroma pictures. Since luma pictures are different from chroma pictures, we trained VRCNN-BN models for luma pictures and chroma pictures, respectively. When inputting a decoded picture in case of 4:2:0 color sampling, the decoded picture is separated to a luma picture and two chroma pictures. Then the luma picture and the chroma pictures are respectively subjected a VRCNN-BN-based post-processing model to obtain output pictures. Finally, the output pictures, including an output picture after post-processing of luma picture and two output pictures after post-processing of chroma pictures, are reconstituted to the post-processing picture in case of 4:2:0 color sampling, which has the same size as the decoded picture.

The proposed coding scheme is shown in Fig 3. Our post-processing algorithm is a decoder-end method for decoded videos. Based on the existing lossy video coding standard, such as HEVC, we improve the coding efficiency at the decoder end.

IV. EXPERIMENTS AND DISCUSSION

In this section, we propose the process of training and testing for VRCNN-BN. The experimental results of VRCNN-BN

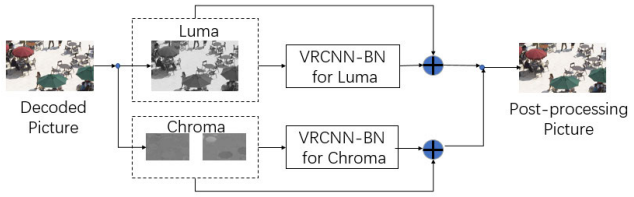


FIGURE 2. The process of VRCNN-BN-based post-processing for luma pictures and chroma pictures.

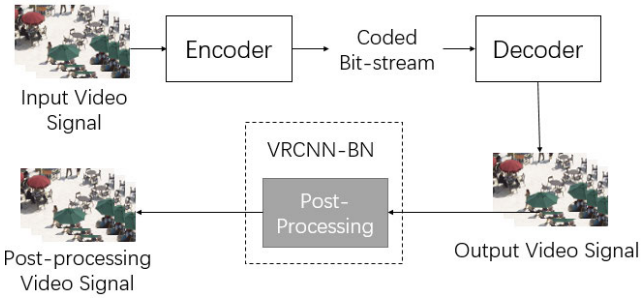


FIGURE 3. VRCNN-BN-based post-processing in coding scheme.

will be compared with HEVC baseline and existing similar algorithms.

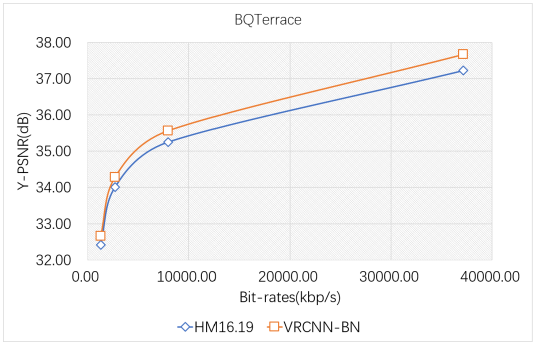
A. TRAINING MODELS

The input of model is the frame Y_n, $n \in (1, \ldots, N)$, from the compressed video Y. The output of model is $F(Y_n \mid \Theta)$, $n \in (1, \ldots, N)$, where Θ is the whole parameters set of VRCNN-BN, including convolution layers and BN layers. And these parameters will be updated at the process of back-propagation. The ground truth of model is the frame X_n, $n \in (1, \ldots, N)$, from the origin video X. The goal of training is to minimize the following loss function:

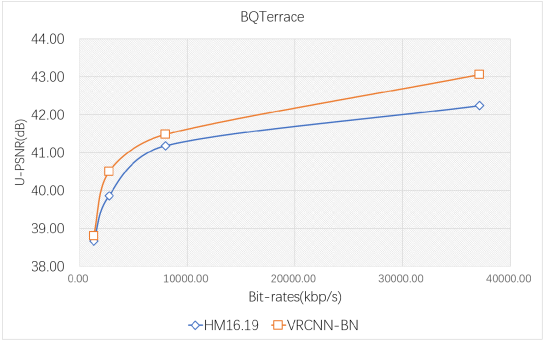
$$L(\Theta) = \frac{1}{N} \sum_{n=1}^{N} \|F(Y_n \mid \Theta) - X_n\|^2 \tag{6}$$

In the experiment, we use the deep learning platform Keras [30] with Tensor Flow backend [31]. During the process of training, the Initialization method of weights is Xavier [32]; the mini-batch size is 120; the optimizer in training is Adam [14], where the learning rate α is set to 0.001, the momentum parameters β_1, β_2 are respectively set to 0.9 and 0.99, and the Infinitesimal itemε is set to 10^{-8}, which is used to avoid errors in learning rate divided by zero, when updating parameters inVRCNN-BN.

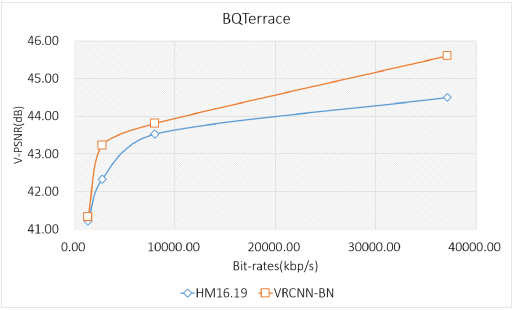
In order to obtain training data, we compressed the sequences under of the common test conditions [9]. All the sequences are compressed by HM16.19 encoder, which is the newest version of HEVC reference software, used four QP settings (22, 27, 32, 37) and four configurations (AI, RA, LDB, LDP). The bit-stream files of compressed sequences are decoded by HM16.19 decoder. In order to extend training data, for each QP, we randomly select frames from compressed sequences, mixing the compressed videos at four configurations, and origin sequences. Then the selected



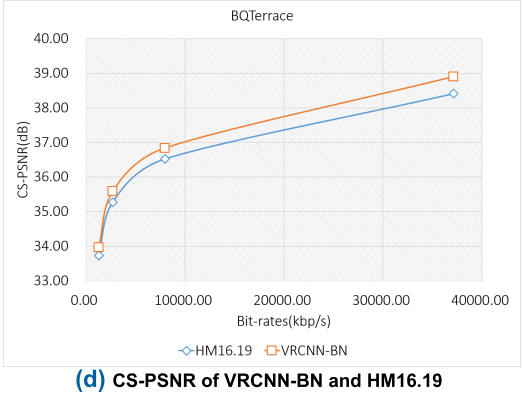
(a) Y-PSNR of VRCNN-BN and HM16.19



(b) U-PSNR of VRCNN-BN and HM16.19



(c) V-PSNR of VRCNN-BN and HM16.19



(d) CS-PSNR of VRCNN-BN and HM16.19

FIGURE 4. Rate distortion curve of BQTerrace 1080p sequence on the case of RA. (a), (b), (c), (b) show the Y-PSNR, U-PSNR, V-PSNR, and CS-PSNR of our approach (VRCNN-BN) and baseline (HM16.19), respectively.

frames are separated into luma pictures and chroma pictures, where luma pictures and chroma pictures are respectively divided into 64×64 sub-pictures on pairs of compressed

FIGURE 5. The fifth frame of Fourpeople and Johnny. (a), (c), (e), (g) show the reduction of blocking artifacts by our approach. And (b), (d), (f), (h) show the reduction of ringing artifacts by our method.

frame and origin frame without overlap. And a pair of corresponding pictures is regarded as a sample. Finally, we select 1394400 luma samples and 679200 chroma samples as training set for each QP.

For each QP, we respectively train the VRCNN-BN models for luma and chroma pictures on our GPU 1080ti. The number of epochs is set to 100. After training, trained models are used to post-process luma pictures and chroma pictures. For each QP, the model for luma pictures are used for processing Y samples and the model for chroma pictures are used for U samples and V samples at the decoder end. There are 8 trained models for 4 QP settings.

B. TESTING MODELS AND COMPARING WITH HEVC BASELINE

For videos in the case of 4:2:0 color sampling, there are Y-PSNR, U-PSNR and V-PSNR for each channel. In order to comprehensively consider the impact of the three channels on video quality, the color-sensitivity-based combined PSNR (CS-PSNR) [33][34] is utilized to evaluate the results of experiment. As described in (7):

$$\begin{aligned} &\mathrm{CS}-\mathrm{PSNR} \\ &\quad = -10\log_{10}\left(\frac{\mathrm{P_Y}}{10^{\frac{\mathrm{Y-PSNR}}{10}}}+\frac{\mathrm{P_U}}{10^{\frac{\mathrm{U-PSNR}}{10}}}+\frac{\mathrm{P_V}}{10^{\frac{\mathrm{V-PSNR}}{10}}}\right) \end{aligned} \tag{7}$$

where $\mathrm{P_Y} = 0.685$, $\mathrm{P_U} = 0.137$, $\mathrm{P_V} = 0.178$.

In (7), the $\mathrm{P_Y}$, $\mathrm{P_U}$, P_V are the percentages of three channels, respectively. The percentages are calculated from the results of a specific experiment. Since we have calculated the PSNR data, the BD-rate of our post-processing algorithm compared to HEVC baseline is obtained from a good interpolation curve through 4 data points [35], including QP22, QP27, QP32, and QP37. With the increase of QP value, PSNR and bit-rates decrease gradually. The quality of videos in low

TABLE 2. The BD-rate results of our VRCNN-BN compared to HEVC baseline on the case of RA and AI.

Sequences		BD-rate							
		RA				AI			
		Y (%)	U (%)	V (%)	CS (%)	Y (%)	U (%)	V (%)	CS (%)
Class A	PeopleOnStreet	-8.2%	-19.5%	-26.4%	-8.9%	-9.3%	-11.8%	-14.1%	-9.4%
	Traffic	-12.1%	-18.9%	-19.4%	-13.0%	-8.7%	-16.5%	-19.3%	-9.8%
Class B	Kimono	-10.2%	-29.0%	-24.9%	-11.9%	-11.8%	-18.1%	-18.1%	-13.2%
	ParkScene	-10.0%	-19.6%	-18.0%	-11.0%	-8.7%	-13.9%	-15.8%	-9.4%
	Cactus	-12.9%	-30.9%	-21.9%	-14.3%	-10.5%	-12.2%	-17.5%	-11.2%
	BasketballDrive	-14.4%	-24.9%	-17.5%	-15.0%	**-14.5%**	-16.8%	-17.4%	**-14.9%**
	BQTerrace	**-19.6%**	**-32.5%**	**-39.1%**	**-20.8%**	-7.7%	**-19.6%**	**-27.2%**	-8.5%
Class C	BasketballDrill	-7.6%	-11.6%	-10.3%	-8.0%	-8.5%	-8.0%	-8.6%	-8.5%
	BQMall	-8.2%	-18.0%	-15.3%	-9.0%	-7.5%	-13.4%	-13.2%	-8.1%
	PartyScene	-2.8%	-9.5%	-8.7%	-3.4%	-4.0%	-5.8%	-7.5%	-4.3%
	RaceHorses	-7.7%	-15.3%	-19.8%	-8.6%	-6.7%	-8.0%	-12.1%	-7.1%
Class D	BasketballPass	-6.6%	-13.0%	-9.6%	-7.0%	-7.4%	-9.0%	-7.3%	-7.4%
	BQSquare	-2.2%	-16.5%	-16.6%	-3.0%	-3.4%	-12.0%	-12.0%	-3.8%
	BlowingBubbles	-3.8%	-10.8%	-8.7%	-4.4%	-4.3%	-7.6%	-7.8%	-4.6%
	RaceHorses	-6.6%	-12.4%	-15.0%	-7.3%	-6.6%	-8.0%	-10.6%	-6.9%
Class E	FourPeople	-11.1%	-19.1%	-17.9%	-11.9%	-9.8%	-16.4%	-17.1%	-10.5%
	Johnny	-14.1%	-18.8%	-18.8%	-14.6%	-10.8%	-18.2%	-17.8%	-11.5%
	KristenAndSara	-11.8%	-17.9%	-20.7%	-12.7%	-9.6%	-16.4%	-18.2%	-10.4%
Class Summary	Class A	-10.2%	-19.2%	-22.9%	-11.0%	-9.0%	-14.2%	-16.7%	-9.6%
	Class B	-13.5%	-27.4%	-24.2%	-14.6%	-10.6%	-16.1%	-19.2%	-11.4%
	Class C	-6.6%	-13.6%	-13.5%	-7.2%	-6.7%	-8.8%	-10.3%	-7.0%
	Class D	-4.8%	-13.2%	-12.5%	-5.4%	-5.4%	-9.2%	-9.4%	-5.7%
	Class E	-12.3%	-18.6%	-19.1%	-13.1%	-10.1%	-17.0%	-17.7%	-10.8%
	Overall	**-9.4%**	**-18.8%**	**-18.3%**	**-10.3%**	**-8.3%**	**-12.9%**	**-14.5%**	**-8.9%**

bit-rates is worse than the videos in high bit-rates, and they are different in PSNR. As shown in Table 2 and Table 3, we test 18 sequences in common HM test conditions [9], from class A to class E. Compared to HM16.19, our approach achieves an average BD-rate (Y-PSNR) reduction of 9.4%, 8.3%, 12.0% and 11.3% for RA, AI, LDP and LDB configurations, respectively. Furtherly, the BD-rate results of CS-PSNR subsume the BD-rate results of Y-PSNR, U-PSNR, and V- PSNR. Compared to HM16.19, our approach achieves average BD-rate reduction, on CS-PSNR, of 10.3%, 8.9%, 13.1% and 11.8% for RA, AI, LDP and LDB configurations, respectively. In Table 4, the average CS-PSNR results of our approach (proposal) is higher than HEVC baseline (anchor) for each QP. The experimental results show that our method can effectively improve the objective quality of videos.

Among the results of all sequences, the results of BQTerrace are especially good. As shown in Fig 4, we test the Y-PSNR, U-PSNR, V-PSNR, and CS-PSNR of our approach and HM16.19 on BQTerrace used QPs (22, 27, 32, 37). Compared to HM16.19, the four-QP-based BD-rate reduction of our approach is 19.6%, 32.5%, 39.1%, and 20.8% on Y-PSNR, U-PSNR, V-PSNR and CS-PSNR, respectively. On the high bit-rates, our approach gets higher quality videos. Especially in the case of QP = 22, V-PSNR of our approach is 1 dB higher than baseline (HM16.19).

As shown in Fig 5, the images obtained by our approach have reduced more artifacts than HVEC baseline. In Fig 5 (a), (c), (e), and (g), the red blocks show the blocking artifacts after video coding. Compared to the original video signal, the HEVC baseline creates a boundary line at the left of face in the red blocks. Our approach weakens this boundary line and the CS-PSNR result of our approach is 0.73 dB higher than HM16.19 on the fifth frame of Fourpeople. In Fig 5 (b), (d), (f), (h), the ringing artifacts are obvious in the blue background area. Our approach reduces the ringing artifacts in Fig 5 (h) and the CS-PSNR result of our approach is 0.66 dB higher than HEVC baseline on the fifth frame of Johnny.

C. COMPARING WITH OTHER METHODS

In this section, we compared our approach with other post-processing algorithms. The video coding efficiency of each approach is evaluated by BD-rate. The results are summarized in Table 5, 6, and 7.

Since VRCNN is a post-processing algorithm for intra HEVC coding according to [17], we list the BD-rate results of our approach and VRCNN for AI configuration in Table 5. On each sequence class, the average BD-rate (Y-PSNR, U-PSNR, and V-PSNR) reduction of our approach is 3.7%, 8.2%, and 9% more than VRCNN in [17], respectively. In contrast to VRCNN, our approach adds BN layers to improve the original VRCNN at the decoder end. During the process of training, the BN layers allow us to use much higher learning rates and fewer training steps [8]. Thus, our model

TABLE 3. The BD-rate results of our VRCNN-BN compared to HEVC baseline on the case of LDP and LDB.

Sequences		BD-rate							
		LDP				LDB			
		Y (%)	U (%)	V (%)	CS (%)	Y (%)	U (%)	V (%)	CS (%)
Class A	PeopleOnStreet	-10.2%	-23.0%	-31.4%	-11.3%	-9.2%	-17.9%	-26.5%	-9.9%
	Traffic	-15.3%	-23.9%	-27.7%	-16.5%	-13.9%	-18.9%	-21.6%	-14.6%
Class B	Kimono	-14.4%	-30.6%	-27.6%	-16.1%	-12.5%	-26.3%	-21.7%	-13.8%
	ParkScene	-11.9%	-22.5%	-23.5%	-13.0%	-11.1%	-19.2%	-18.7%	-11.9%
	Cactus	-16.4%	-36.0%	-28.2%	-18.0%	-14.8%	-25.2%	-19.3%	-15.4%
	BasketballDrive	-16.5%	-31.4%	-22.1%	-17.4%	-15.9%	-26.5%	-14.3%	-16.1%
	BQTerrace	-20.6%	**-47.0%**	**-58.6%**	**-22.9%**	**-19.6%**	**-36.4%**	**-47.5%**	**-21.1%**
Class C	BasketballDrill	-8.8%	-10.8%	-12.0%	-9.2%	-9.0%	-9.0%	-6.9%	-8.8%
	BQMall	-9.7%	-21.3%	-19.6%	-10.6%	-9.3%	-17.1%	-13.9%	-9.8%
	PartyScene	-4.5%	-11.6%	-11.4%	-5.0%	-4.5%	-8.4%	-7.4%	-4.8%
	RaceHorses	-8.9%	-16.4%	-22.8%	-9.8%	-8.7%	-13.6%	-16.9%	-9.2%
Class D	BasketballPass	-8.0%	-14.7%	-9.2%	-8.3%	-7.8%	-11.9%	-6.6%	-7.9%
	BQSquare	-3.7%	-28.3%	-28.8%	-4.7%	-4.4%	-23.6%	-22.0%	-5.2%
	BlowingBubbles	-5.6%	-13.2%	-11.7%	-6.2%	-5.6%	-10.5%	-7.6%	-5.9%
	RaceHorses	-7.6%	-12.4%	-15.5%	-8.3%	-7.4%	-10.7%	-11.4%	-7.7%
Class E	FourPeople	-15.6%	-31.1%	-29.5%	-16.9%	-14.4%	-24.5%	-20.9%	-15.1%
	Johnny	**-21.0%**	-32.9%	-32.4%	-22.2%	-18.7%	-20.8%	-19.0%	-18.8%
	KristenAndSara	-17.7%	-30.6%	-32.3%	-19.1%	-16.2%	-22.1%	-22.2%	-16.8%
Class Summary	Class A	-12.8%	-23.5%	-29.6%	-13.9%	-11.6%	-18.4%	-24.1%	-12.2%
	Class B	-16.0%	-33.5%	-32.0%	-17.5%	-14.8%	-26.7%	-24.3%	-15.6%
	Class C	-8.0%	-15.0%	-16.4%	-8.7%	-7.9%	-12.0%	-11.3%	-8.1%
	Class D	-6.2%	-17.2%	-16.3%	-6.9%	-6.3%	-14.2%	-11.9%	-6.7%
	Class E	-18.1%	-31.5%	-31.4%	-19.4%	-16.4%	-22.5%	-20.7%	-16.9%
	Overall	**-12.0%**	**-24.3%**	**-24.7%**	**-13.1**	**-11.3%**	**-19.0%**	**-18.0%**	**-11.8%**

TABLE 4. The average CS-PSNR results of our VRCNN-BN compared to HEVC baseline on the case of RA, AI, LDP and LDB.

Sequences		CS-PSNR (dB)							
		QP22		QP27		QP32		QP37	
		anchor	proposal	anchor	proposal	anchor	proposal	anchor	proposal
Class A	PeopleOnStreet	42.05	**42.80**	38.94	**39.43**	36.12	**36.55**	33.43	**33.75**
	Traffic	42.36	**43.10**	39.71	**40.22**	37.25	**37.68**	34.74	**35.05**
Class B	Kimono	42.52	**43.24**	40.60	**41.08**	38.46	**38.90**	36.20	**36.48**
	ParkScene	41.04	**41.67**	38.32	**38.73**	35.75	**36.10**	33.33	**33.57**
	Cactus	39.78	**40.33**	37.83	**38.22**	35.93	**36.29**	33.81	**34.06**
	BasketballDrive	40.79	**41.41**	38.79	**39.23**	36.97	**37.35**	35.01	**35.28**
	BQTerrace	39.90	**40.46**	36.88	**37.25**	35.11	**35.45**	33.11	**33.37**
Class C	BasketballDrill	41.44	**42.05**	38.27	**38.69**	35.51	**35.81**	33.09	**33.32**
	BQMall	41.42	**42.07**	38.71	**39.15**	35.98	**36.33**	33.25	**33.49**
	PartyScene	39.76	**40.23**	36.07	**36.30**	32.89	**33.03**	29.84	**29.95**
	RaceHorses	40.78	**41.42**	37.43	**37.83**	34.58	**34.87**	31.86	**32.07**
Class D	BasketballPass	41.95	**42.64**	38.24	**38.69**	35.08	**35.40**	32.28	**32.51**
	BQSquare	39.98	**40.44**	36.28	**36.47**	33.28	**33.41**	30.39	**30.44**
	BlowingBubbles	39.55	**40.03**	35.93	**36.19**	32.69	**32.87**	29.74	**29.88**
	RaceHorses	40.85	**41.51**	37.08	**37.51**	33.82	**34.12**	31.03	**31.26**
Class E	FourPeople	43.86	**44.66**	41.69	**42.28**	39.23	**39.74**	36.47	**36.86**
	Johnny	44.29	**45.04**	42.55	**43.05**	40.53	**40.98**	38.18	**38.51**
	KristenAndSara	44.49	**45.27**	42.52	**43.05**	40.25	**40.74**	37.71	**38.06**

is trained on a big data set, including 1394400 luma samples and 679200 chroma samples, and performers better on test sequences.

We list the average BD-rate results of our approach, DCAD [18], and Residual-VRN [20], [21] for RA, AI, LDP, and LDB configurations in Table 6. For each configuration, since DCAD and Residual-VRN are post-processing algorithms for luma pictures, the average BD-rate reduction on Y-PSNR of our approach is 4.55% more than DCAD and 2.37% more than Residual-VRN. Compared to DCAD and Residual-VRN, our approach uses different sizes of convolution kernels in one layer, which allows us to simplify the network and performer better in post-processing.

Table 7 shows the BD-rate results of our approach and VDSR in [19] on cif sequences [26]. According to [19], we take the x265 platform as encoder, which is same as

TABLE 5. The BD-rate performance of vrcnn-BN comparing with Vrcnn [17] FOR hevc intra coding.

Testset		Average BD-rate		
		Y	U	V
Class A	VRCNN	-3.5%	-3.7%	-3.6%
	VRCNN-BN	**-9.0%**	**-14.2%**	**-16.7%**
Class B	VRCNN	-3.3%	-3.2%	-3.7%
	VRCNN-BN	**-10.6%**	**-16.1%**	**-19.2%**
Class C	VRCNN	-5.0%	-5.5%	-6.9%
	VRCNN-BN	**-6.7%**	**-8.8%**	**-10.3%**
Class D	VRCNN	-5.4%	-6.4%	-8.1%
	VRCNN-BN	**-5.4%**	**-9.2%**	**-9.4%**
Class E	VRCNN	-6.5%	-5.5%	-5.6%
	VRCNN-BN	**-10.1%**	**-17.0%**	**-17.7%**
Overall	VRCNN	-4.6%	-4.7%	-5.5%
	VRCNN-BN	**-8.3%**	**-12.9%**	**-14.5%**

TABLE 6. The average BD-rate performance of VRCNN-BN comparing with Residual-VRN [20], [21] and DCAD [18] in the case of RA, AI, LDP and LDB for RA, AI, LDP and LDB configurantions.

Configurantions	Average BD-rate		
	DCAD	Residual-VRN	VRCNN-BN
RA	-6.10%	-7.30%	**-9.40%**
AI	-5.00%	-7.41%	**-8.30%**
LDP	-6.40%	-9.40%	**-12.00%**
LDB	-5.30%	-7.40%	**-11.30%**
avarage	-5.70%	-7.88%	**-10.25%**

TABLE 7. The BD-rate performance of vrcnn-BN comparing with VDSR [19] on cif sequences.

Size	Sequences	Average BD-rate	
		VDSR	VRCNN-BN
352x288	Football_cif	-1.9%	**-7.0%**
	Foreman_cif	-2.4%	**-7.4%**
	Flower_cif	-1.1%	**-2.2%**
Overall		-1.8%	**-5.5%**

their approach. On each cif sequence, because VDSR is applied to luma pictures, the BD-rate reduction on Y-PSNR of our approach is 3.7% more than VDSR. As shown in Table 7, the results of experiment indicate that our approach is still valid for the videos with low bitrates and poor quality. Compared to the videos with high bitrates, the neural network cannot extract enough information to reconstruct the image.

V. CONCLUSION

In this paper, we present a CNN-based post-processing algorithm for luma pictures and chroma pictures at the decoder end. The proposed VRCNN-BN inspired by VRCNN, VSDR and Residual-VRN. Our approach is better than pervious methods in achieving higher BD-rate reduction, less artifacts, and better video quality. Meanwhile, our proposed algorithm is applicable not only to luma pictures but also to chroma pictures. In the usual case of 4:2:0 color sampling, we test the BD-rate of CS-PSNR combined the Y-SNR, U-PSNR and V-PSNR. Compared to HEVC baseline, we gain the BD-rate reduction of 10.3%, 8.9%, 13.1%, and 11.8% on CS-PSNR for RA, AI, LDP, and LDB configurations.

In the future, we plan to apply our post-processing method to in-loop filters at the encoder end. Our method further improves the coding efficiency not only by improving video quality at the decoder end but also by reducing bit-rates at the encoder end.

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] *IEEE Standard for Second-Generation IEEE 1857 Video Coding*, Standard 1857.4, Aug. 2019.
[3] C. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, "Sample adaptive offset for HEVC," in *Proc. IEEE 13th Int. Workshop Multimedia Signal Process.*, Hangzhou, China, Oct. 2011, pp. 1–5.
[4] A. Norkin, "HEVC Deblocking Filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.
[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, CA, USA, May 2015.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[8] Ioffe, Sergey, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2015.
[9] F. Bossen, *Common HM Test Conditions and Software Reference Configurations*, document JCTVC-K1100, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11th Meeting, Shanghai, China, Oct. 2012.
[10] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. H. Hong, M.-S. Cheon, N. Shlyakhov, K. McCann, T. Davies, and J.-P. Park, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.
[11] G. Zhai, W. Zhang, X. Yang, W. Lin, and Y. Xu, "Efficient deblocking with coefficient regularization, shape-adaptive filtering, and quantization constraint," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 735–745, Aug. 2008.
[12] C. H. Yeh, S. J. Jiang, T. F. Ku, M. J. Chen, and J. A. Jhu, "Post-processing deblocking filter algorithm for various video decoders," *IET Image Process.*, vol. 6, no. 5, pp. 534–547, 2012.
[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, May 2015.
[15] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *Proc. IEEE 12th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Bordeaux, France, Jul. 2016, pp. 1–5.
[16] 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 Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 26–30.

[17] D. Yuanying, L. Dong, and W. Feng, "A convolutional neural network approach for post-processing in HEVC intra coding," in *Proc. Int. Conf. Multimedia Modeling*. Cham, Switzerland: Springer, 2017, pp. 28–39.
[18] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2017, pp. 410–419.
[19] C. Li, L. Song, R. Xie, and W. Zhang, "CNN based post-processing to improve HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 4577–4580.
[20] L. Ma, Y. Tian, and T. Huang, "Residual-based video restoration for HEVC intra coding," in *Proc. IEEE 4th Int. Conf. Multimedia Big Data (BigMM)*, Xi'an, China, Sep. 2018, pp. 1–7.
[21] L. Ma, Y. Tian, P. Xing, and T. Huang, "Residual-based post-processing for HEVC," *IEEE Multimedia*, vol. 26, no. 4, pp. 67–79, Oct./Dec. 2019.
[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*. Zurich, Switzerland: Springer, Sep. 2014.
[23] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 1–17, Feb. 1992.
[24] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 576–584.
[25] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1646–1654.
[26] Grand Challenge ICIP. (2017). *Grand Challenge on the Use of Image Restoration for Video Coding Efficiency Improvement*. [Online]. Available: https://storage.googleapis. com/icip-2017/index.html
[27] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "CNN-based detection of generic contrast adjustment with JPEG post-processing," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 3803–3807.
[28] H. Ma, D. Liu, R. Xiong, and F. Wu, "A CNN-based image compression scheme compatible with JPEG-2000," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 704–708.
[29] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Process., Image Commun.*., vol. 17, no. 1, pp. 3–48, Jan. 2002.
[30] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/ keras-team/keras
[31] M. Abadi, "Tensorflow: A system for large-scale machine learning," in *Proc. Symp. Operating Syst. Design Implement.*, 2016, pp. 265–283.
[32] G. Xavier and Y. Bengio, "Understanding the difficulty of training deep feed forward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[33] X. Shang, G. Wang, H. Zhao, J. Liang, C. Wu, and C. Lin, "A new combined PSNR for objective video quality assessment," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Hong Kong, Jul. 2017, pp. 811–816.
[34] X. Shang, J. Liang, G. Wang, H. Zhao, C. Wu, and C. Lin, "Color-sensitivity-based combined PSNR for objective video quality assessment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1239–1250, May 2019.
[35] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, ITU-T SG16, Q6, Apr. 2001.

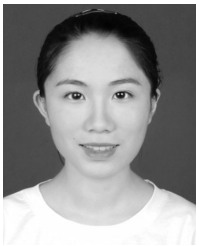
HAIWU ZHAO received the B.S., M.E., and Ph.D. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1996, 1999, and 2003, respectively. From 2003 to 2005, he was a Postdoctoral Associate with East China Normal University, Shanghai, China. He is currently an Associate Professor with the School of Communication and Information Engineering, Shanghai University. His research interests include image processing, video coding, and machine learning.

MING HE received the B.S. degree in communication engineering from Shanghai University, Shanghai, China, in 2017, where he is currently pursuing the M.E. degree with the School of Communication and Information Engineering. His research interests include video coding and deep learning.

GUOWEI TENG received the M.E. degree in electrical circuit and system from the University of Chinese Academy of Sciences, Jilin, China, and the Ph.D. degree in communication and information system from Shanghai University, Shanghai, China, where he is currently a Professor with the School of Communication and Information Engineering. His research interests include video coding and image resolution.

XIWU SHANG received the B.E. degree from Hubei University, Hubei, China, in 2010, and the M.E. and Ph.D. degrees from Shanghai University, Shanghai, China, in 2014 and 2018, respectively. From 2016 to 2017, he was a joint Ph.D. Student with the School of Engineering Science, Simon Fraser University, Canada. Since 2018, he has been with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science. His research interests include video coding, quality assessment, and deep learning.

GUOZHONG WANG received the M.E. degree in computer application from the Nanjing University of Science and Technology, Nanjing, China, in 1998, and the Ph.D. degree in system integration from East China Normal University, Shanghai, China, in 2006. He is currently a Professor with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science. His research interests include image processing, video coding, and video communication.



YIYAN FENG received the B.S. degree in information engineering from Shanghai University, Shanghai, China, in 2018, where she is currently pursuing the Ph.D. degree with the School of Communication and Information Engineering. Her research interests include video coding and quality assessment.