Residual-Based Video Restoration for HEVC Intra Coding

Li Ma[1,2], Yonghong Tian[#2], Tiejun Huang[2]
[1]*Center for Data Science, AAIS*
[2]*National Engineering Laboratory for Video Technology, School of EECS*
Peking University
Beijing, China
[#]The corresponding author is Prof. Yonghong Tian at yhtian@pku.edu.cn

***Abstract*—Inspired by the great success of convolutional neural network (CNN) in computer vision, CNN-based post-processing methods at the decoder side have achieved significant advances in improving the video quality. However, these methods only learn a mapping from the decoded frame to an artifact-free reconstruction, which ignores that the distortion comes from quantization of the prediction residual transform coefficients. In this paper, we propose a residual-based video restoration network (Residual-VRN) to improve the quality of decoded video, in which the coded prediction residual is combined with the prediction frame as the input of the network. Meanwhile, the activation function, the residual learning framework and the loss function can also be optimized to achieve better quality enhancement. Experimental results show that the proposed Residual-VRN leads to an average 7.41% BD-rate reduction compared to the HEVC intra coding baseline, outperforming the conventional CNN-based video restoration algorithms. In deeper CNN architectures, our method achieves 8.0%, 9.0% and 11.1% BD-rate savings, much higher than the conventional CNN-based post-processing methods.**

***Index Terms*—Decoder-side enhancement, Artifact reduction, Convolutional neural network (CNN), High Efficiency Video Coding (HEVC), Intra coding, Post-processing**

I. Introduction

Lossy codec algorithms for images and videos, such as the well-known JPEG [1] and High Efficiency Video Coding (HEVC) [2] have achieved an impressive coding efficiency. As a compromise to lower the bit-rate, the distortions are ineluctably induced in these codecs, especially at lower bit-rates. A traditional way to improve the quality of reconstructed video or pictures is to apply in-loop filters to the encoder. Recently, different from the traditional way, post-processing methods based on deep convolutional neural network (CNN) have achieved significant advances [3], [4]. These CNN models learned a non-linear mapping from the decoded video to original frame. However, some of the modules in these CNN-based video restoration models do not work efficiently in video restoration. On the other hand, the conventional video restoration models only take the reconstruction frame as input which ignores that the coded prediction residual has great potential to improve the video quality. As we know, the reconstructed frame can be obtained by summing the prediction frame with the coded prediction residual frame. Due to the block-based transform and quantization at the encoder side, the coded prediction residual information contains some objects edge or contour in the frame which can be combined into the post-processing procedure.

It needs to be specifically stated that the residuals in this paper have different meanings in different places. The prediction residual, coded prediction residual and residual signal represents residual in HEVC, but the global residual-learning, local residual-learning and residual block represent residual-learning framework in Deep Learning.

In this paper, we firstly analyze the structure of existing CNN models and improve the activation function, residual learning framework and loss function. More precisely, we propose a Double-Channel Rectified Linear Units (DC-ReLU) as our CNN's activation function rather than ReLU and use a mixed loss function rather than $\mathfrak{L}_2$ loss. Finally, we take the coded prediction residual into account, then propose a residual-based video restoration network (Residual-VRN) to automatically enhance the quality of decoded frame. Experimental results show that Residual-VRN achieves a higher bit-rate reduction than previous decoded-frame-based network that we call the conventional video restoration networks. In order to demonstrate the superiority of our residual based approach, we change the structure of the conventional VRNs to the same with Residual-VRN. We call the network that have been changed improved video restoration network (Improved-VRN). In order to exploit the potential of Residual-VRN, we deepen it by stacking the residual blocks to construct 16-residual-block, 32-residual-block, 64-residual-block networks. We name the three network Deep Residual-VRN-16 (DR-VRN-16), Deep Residual-VRN-32 (DR-VRN-32) and Deep Residual-VRN-64 (DR-VRN-64). Some method to accelerate the training of CNN are used, such as learning rate decay and residue learning technique [5]. Moreover, in order to demonstrate the robustness of our network, we trained the network with a collection of natural images and tested the network with the standard video sequences. Our models can be adopted as post-processing to replace deblocking and SAO, and it require no additional bit.

It should be emphasized that, in this paper, we mainly focus on improving coding efficiency by combining the HEVC and deep learning at the decoder end. The remainder of this paper is organized as follows: Section 2 reviews the related

978-1-5386-5321-0/18/$31.00

works. We analyze the deficiency of the conventional networks and explain why we using coded prediction residual frame in Section 3. In Section 4, we introduce Double-ReLU activation function and $\tilde{\mathfrak{L}_{\text{mix}}}$, and propose Residual-VRN. Experimental details are demonstrated and extensive experimental results are reported in Section 5. Conclusions are drawn in Section 6.

The main contributions of this paper are as below:

- We take the coded prediction residual into account, and propose a residual-based video restoration network (Residual-VRN).
- We propose DC-ReLU as our CNN's activation function rather than ReLU and use a mixed loss function rather than $\mathfrak{L}_2$ loss.
- We deepen our Residual-VRN to exploit the potential of it, and Residual-VRN hold greater potential than decoded-frame-based approach.

II. RELATED WORKS

A. HEVC Intra Coding

In HEVC intra coding, the residual signal of intra-prediction is transformed by a linear spatial transform. The transform coefficients are then scaled, quantized, entropy coded, and transmitted together with the prediction information. The encoder duplicates the decoder processing loop such that both will generate identical predictions for subsequent data. Therefore, the quantized transform coefficients are constructed by inverse scaling and are then inverse transformed to duplicate the decoded approximation of the residual signal. The decoded prediction residual frame is then added to the prediction, and the result of that addition may be fed into one or two loop filters. We combine the residual signal in frame level and formulate this process as:

$$R = P + \tilde{\mathcal{R}} = P + T^{-1}Q^{+}QT(O - P) \tag{1}$$

where the O is the original frame; the P represents the prediction frame; the $\tilde{\mathcal{R}}$ is the approximation of the prediction residual frame in decoding processing, i.e. coded prediction residual frame; the R is the decoded frame before utilizing loop filters. $Q(\cdot)$ and $Q^{+}(\cdot)$ represent the quantization and de-quantization; $T(\cdot)$ and $T^{-1}(\cdot)$ represent the transform and inverse transform. As we know, HEVC design supports a total of 35 intra prediction modes [6]. In order to choose the most efficient mode, all candidates (35 modes) are evaluated with respect to the following cost function: $C = D + \lambda \cdot \Re$, where the D and $\Re$ are the distortion and bitrate for the sequence; and λ is the Lagrange multiplier. In frame level, if the other encoding parameters are fixed, we have

$$\begin{aligned} m_{opt} &= \underset{m,param}{argmin} C \\ &= \underset{m,param=param^{*}}{argmin} (D_F + \lambda\Re(F)) \\ &= \underset{m,param=param^{*}}{argmin} (D(\mathcal{R}) + \lambda\Re(\mathcal{R} + P + param)) \\ &= \underset{m,param=param^{*}}{argmin} (D(T^{-1}Q^{+}QT(\mathcal{R})) \\ &+ \lambda\Re(T^{-1}Q^{+}QT(\mathcal{R}) + P + param)) \end{aligned} \tag{2}$$

where the m_{opt} is the most efficient intra-prediction modes, and $m_{opt} \in \{0, 1, 2, ..., 34\}^{q}$, where $q = \left\lceil \frac{h}{minsize_{TU}} \right\rceil \times \left\lceil \frac{w}{minsize_{TU}} \right\rceil$, h, w denote the height and width of sequence. , $minsize_{TU}$ denote the smallest size of the prediction block. The $para$ and $param^{*}$ are the other encoding parameters and the latter is fixed. The D_F and $D(\mathcal{R})$ are the distortion of the frame and the prediction residual frame. The $D(\mathcal{R})$ is equal to the D_F, because the distortion comes from the quantization of the residual signal's transform coefficients. The $\Re(X)$ represent the bitrates of X , and the P represents the predicted frame of encoding processing. Due to $\{0, 1, 2, ..., 34\}$ is a finite set, then we have:

$$\tilde{\mathcal{R}} = \hat{\mathcal{R}}_{m_{opt}} = G(P, \mathcal{R}, para) \tag{3}$$

$$(P, \tilde{\mathcal{R}}) = H(O, para) \tag{4}$$

G and H are defined to make it easier to describe what the Residual-VRN did in the next section.

B. In-Loop Filters in HEVC

In HEVC, the state-of-the-art video coding standard, there are two post-processing techniques for artifact reduction, namely deblocking [7] and sample adaptive offset (SAO) [8]. There are two major differences between them. Firstly, deblocking is specifically designed to reduce blocking artifacts, but SAO is designed for general compression artifacts reduction. Secondly, deblocking does not require any additional bit, but SAO requires to transmit some additional bits for signaling the offset values. Both techniques contribute to the improvement of the visual quality of reconstructed video equivalently achieve improving the coding efficiency.

C. Convolution Neural Network based Methods

Recently, convolutional neural network (CNN) achieved great success in high-level computer vision tasks such as image classification [9] and object detection [10]. Inspired by the success, it was also proposed to utilize CNN for low-level computer vision tasks such as super-resolution [11] [12], edge detection [13] and image restoration [14].

More recently, Dong et al. proposed an artifact reduction CNN (AR-CNN) [15] approach for reducing artifacts in JEPG compressed images, and reported to achieve more than 1 dB improvement over JPEG images. Wang et al. [16] investigated another network structure for JPEG artifact reduction. Park and Kim [17] proposed to utilize the CNN network to replace the deblocking or SAO in HEVC, and reported achieving bit-rate reduction. However, the results in [17] were achieved by training a network with several frames of a video sequence and then testing the network with the same sequence, which cannot reveal the generalizability of the trained network. Dai and Liu [3] proposed a Variable-filter-size Residue-learning CNN (VRCNN) mainly for artifact reduction in HEVC intra coding, and achieve on average 4.6% bit-rate reduction compared to deblocking and SAO in HEVC baseline. Wang and Chen [4] proposed a Deep CNN-based AutoDecoder (DCAD). In their training data selection, they leverage the

TU size information recorded in the bitstreams to select the patches in the training set, which makes the distribution of TU uniform. However, all of their work ignore the residual signal in encoding process where the distortion come from. This conventional video restoration networks can be formulated as $\min_{\theta} \|\mathcal{F}(R|\theta) - O\|^2$, where O represents the original frame, R represents the decoded frame, and $\mathcal{F}(R|\theta)$ denotes their networks.

III. Analysis on Video Restoration Network

In this section, we analyze the deficiency of the conventional VRNs versatilely, and then introduce why we use coded prediction residual frame.

A. Structure of VRN

1) Activation function: In the conventional VRNs, they all utilize ReLU as activation function, which extensively used in deep learning. Glorot et al. [18] had explained that ReLU allows a network to obtain sparse representations easily. However, ReLU still has some limitations. Precisely, ReLU ignores the response of the negative phase of the feature maps,and leads to the loss of some improtant details. We thought the negative phase should be considered equally with the positive phase. On the other hand, ReLU is a single-threshold activation function, but Li et al. [19] have proven that the activation with multi-threshold performs better than single-threshold.

2) Residual learning: Although the conventional VRNs utilized residual learning technique, residual learning framework of them is global residual learning. We find that there are several low frequency feature maps in each layer of the conventional VRNs, which is redundant and inefficient for high-level structure information learning.

3) Loss function: Most of existing CNN based low-level vision method use the L_2 norm (5) based reconstruction error usually called MSE as the loss function e.g. [17] [15] [12]. However, the L_2 norm is not robust to outliers and usually leads to results contain noise and ringing artifacts [20] on low-level computer vision tasks.

$$\mathfrak{L}_2(O, \tilde{O}) = \| O - \tilde{O} \|^2 \tag{5}$$

B. Why Using the Coded Prediction Residual Frame

In decoding processes of HEVC, we can get the $\tilde{\mathcal{R}}$ and the P ,and the $\mathcal{R}$ is unknown, so we can't get the O by :$O = P + \mathcal{R}$, where $\mathcal{R}$ denotes the prediction residual frame in encoder side. The HEVC decoding processing can be regarded as:

$$R = P + \tilde{\mathcal{R}} \approx P + \mathcal{R} = O \tag{6}$$

The conventional VRNs treat this processing as:

$$\hat{O} = F(R) = F(P + \tilde{\mathcal{R}}) \approx O \tag{7}$$

We treat it as:

$$\tilde{O} = P + G^+(P, \tilde{\mathcal{R}}, param) \approx O \tag{8}$$

We get $\tilde{O}$ as our enhanced frame. The $G^+(\cdot)$ defined as a function:

$$G^+(P, \tilde{\mathcal{R}}, para) = \{E(O - P)|H(O, para) = (P, \tilde{\mathcal{R}})\} \tag{9}$$

Where $E(O-P)$ represents the expectation of the distribution of $O - P$. We can't formulate (8) as:

$$O \approx P + G^{-1}(P, \tilde{\mathcal{R}}, para) \tag{10}$$

because the $G(\cdot)$ is not invertible. On the other hand, if the $G(\cdot)$ is invertible, we could get the O by handcraft method. Therefore, we define the $G^+(\cdot)$. As the (8) shows, if we want get more approximate O to improve our encoding efficience, the first step is to attempt to get the function $G^+(\cdot)$,which is very hard to the traditional method, but the deep-learning method is qualified for this work.

IV. Method

A. Double-Channels ReLU Activation Function

ReLU ignores the response of the negative phase of the feature maps, which we thought that should be considered equally with the positive phase. In order to overcome the shortcomings of ReLU, we propose a novel multi-threshold activation function, i.e. Double-Channels ReLU (DCReLU) function as our activation function which is defined as

$$DCReLU(x) = [max(x - \eta_1, 0), min(\beta \times x - \eta_2, 0)] \tag{11}$$

where β is a trainable scale parameter initialized with the value of 0.5. η_1 and η_2 are the bias thresholds that are also trainable.

B. Residual Block

For the purpose of reducing the low frequency redundancy in a deep CNN, we utilize the local residual learning in ResNet [5]. The main idea of ResNet is to use a residual learning framework to ease the training of very deep networks. As Fig.1 shows, our residual block's structure can be formulated as:

$$RB(x) = x + U(x) = x + F_2(\sigma_{DC-ReLU}(F_1(x))) \tag{12}$$

where $RB(x)$ is the output of residual block, function $\sigma_{DC-ReLU}(\cdot)$ denotes the Double-Channel ReLUs activation function, $F_1(\cdot)$ and $F_2(\cdot)$ denote the $conv - 3 \times 3 \times 64$ and $conv - 1 \times 1 \times 64$, $U(x)$ is the residual to be learned.

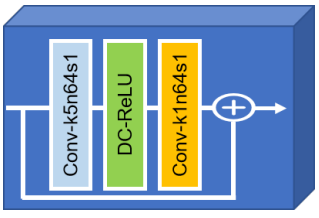

Fig. 1. Residual Block's Structure with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer

C. Mixed Loss Function

In order to overcome the limitation of L_2 norm based reconstruction error, Zhao et al. [20] proposed a mixed loss function:

$$\mathfrak{L}_{\text{mix}} = \alpha \cdot G_\sigma \cdot \mathfrak{L}_1(O, \tilde{O}) + (1-\alpha) \cdot \mathfrak{L}_{MS-SSIM}(O, \tilde{O}) \quad (13)$$

However, their $\mathfrak{L}_{\text{mix}}$ doesn't work on our task. We use an mixed loss function with $\mathfrak{L}_2$ loss:

$$\begin{aligned}\tilde{\mathfrak{L}_{mix}} =& \gamma \cdot (G_\sigma \cdot \mathfrak{L}_1(O, \tilde{O}) + \mathfrak{L}_{MS-SSIM}(O, \tilde{O})) \\ &+ \sqrt{G_\sigma \cdot \mathfrak{L}_2(O, \tilde{O})}\end{aligned} \quad (14)$$

, without using the accelerating trick proposed by Zhao [20]. $\gamma = 0.5$ in our experiment, halved every 50,000 iteration. Then, we validate the effect of the loss function.

D. Architecture of Residual-VRN

As Fig 2 shows, we utilize predicted frame P and coded prediction residual frame $\tilde{\mathcal{R}}$ as the input to Residual-VRN. A normalization layer is applied to reduce the difference of distribution of P and $\tilde{\mathcal{R}}$. After a stock of convolutional layers and residual blocks, the $\tilde{\mathcal{R}}$ and P are add to the result, in order to do the residual learning of the coded prediction residual frame. Residual-VRN can be formulated as:

$$\tilde{O} = F_3(ReLU(F_2(RB^6(F_1(Norm(\tilde{\mathcal{R}}, P)))))) + P + \tilde{\mathcal{R}} \quad (15)$$

where $F_1(\cdot)$ represents a $5 \times 5 \times 64$ convolution layer, $F_2(\cdot)$ represents a $3 \times 3 \times 64$ convolution layer, $F_3(\cdot)$ represents a $3 \times 3 \times 1$ convolution layer, and $Norm(\cdot)$ denotes the normalization layer. In order to compare our residual-based network with decoded-frame-based network fairly, we improve the conventional networks by adjusting their structure to the same as Residual-VRNs. The improved network is named Improved-VRN.

E. Deep Residual-VRN

It should be emphasized that experimental result shows Residual-VRN performs better with less parameters than the conventional networks. For the purpose of exploiting the potential of Residual-VRN, we deepen it by stacking the residual blocks to construct 16-residual-block, 32-residual-block, 64-residual-block networks. We name the three network Deep Residual-VRN-16 (DR-VRN-16), Deep Residual-VRN-32 (DR-VRN-32) and Deep Residual-VRN-64 (DR-VRN-64). On the other hand, we do the same for the conventional VRNs in order to compare the potential with Residual-VRN, and we get Deep Improved-VRN-16 (DI-VRN-16), Deep Improved-VRN-32 (DI-VRN-32) and Deep Improved-VRN-64 (DI-VRN-64).

V. EXPERIMENT

In this section, we evaluate the performance of our models on the standard video sequences. We first briefly introduce the details of our training. Then, we conduct several experiments to investigate the properties our model. Finally, we compare our models with several state-of-the-art methods. For simplisity, in some comparative experiments only the luminance channel is considered for test.

A. Training Details

1) Datasets: We randomly select 10,000 images from MS-COCO [21] and turn them to YUV format. Each YUV is compressed by HEVC intra coding (deblocking and SAO turned off) at four different QPs: 22, 27, 32, 37. For each QP, a separate network is trained out. Only the luminance channel (i.e. Y out of YUV) is considered for training. We choose MS-COCO just because it can supply large size images.

2) Training steps: An original frame O_i, where $i \in \{1, ..., N\}$ indexes each frame, is compressed with HEVC intra coding, with the deblocking and SAO turning off. The predicted frame P_i and the coded prediction residual frame $\tilde{\mathcal{R}}_i$ are regarded as the input to our network. The objective of training Residual-VRN is to minimize the following loss function:

$$\tilde{\theta} = min_\theta \mathfrak{L}_{mix}(N(P, \tilde{\mathcal{R}}, \theta), O) \quad (16)$$

where $N(P, \tilde{\mathcal{R}}, \theta)$ denotes the result of Residual-VRN.

3) Implementation details: We use TensorFlow [22] for training our networks on a NVIDIA Tesla K80 graphical processing unit (GPU). During network training, the weights are initialized using the method in [5]. Training samples are randomly shuffled and the mini-batch size is 32. For each mini-batch, we randomly capture a 200×200 patch from each image in each mini-batch. Adam optimizer [23] with β_1=0.9,β_2=0.999 and $\epsilon = 1e^{-8}$ is used to optimize the parameters. The training rate is initially set to $1e^{-4}$ and halved every 50,000 iterations. The Residual-VRN is trained totally 150,000 iterations.

B. Effect of Loss Function

We trained several Residual-VRN with different loss functions:$\mathfrak{L}_1$, $\mathfrak{L}_2$, $\mathfrak{L}_{mix}$, $\tilde{\mathfrak{L}_{mix}}$, and the other hyper-parameters are same. Table I shows a comparison of results produced by the networks mentioned above. It can be seen that the network trained with $\tilde{\mathfrak{L}_{mix}}$ performs better than the others.

TABLE I
BD-RATE RESULTS OF RESIDUAL-VRN TRAINED BY DIFFERENT LOSS FUNCTION

Loss Function		$\mathfrak{L}_1$	$\mathfrak{L}_2$	$\mathfrak{L}_{mix}$	$\tilde{\mathfrak{L}_{mix}}$
Class B	Kimono	-9.8%	-9.8%	-9.4%	**-9.9%**
	ParkScene	-0.7%	-0.5%	-0.7%	**-1.1%**
	Cactus	-6.1%	-5.6%	-5.8%	**-6.4%**
	BasketballDrive	**-6.2%**	-5.2%	-5.3%	-6.1%
	BQTerrace	-2.9%	-2.3%	-2.5%	**-3.1%**
Class C	BasketballDrill	-10.6%	-9.9%	-10.4%	**-11.5%**
	BQMall	-7.3%	-6.7%	-6.8%	**-7.5%**
	PartyScene	-5.3%	-5.1%	-5.2%	**-5.6%**
	RaceHorses	-2.1%	-2.5%	**-2.8%**	**-2.8%**
Class D	BasketballPass	-9.0%	-8.5%	-8.7%	**-9.4%**
	BQSquare	-7.1%	-6.5%	-6.8%	**-7.6%**
	BlowingBubble	-6.8%	-6.5%	-6.7%	**-7.2%**
	RaceHorses	-10.5%	-10.4%	-10.6%	**-11.0%**
Class E	FourPeople	-9.5%	-9.0%	-9.1%	**-9.7%**
	Johnny	-9.3%	-8.6%	-8.9%	**-9.8%**
	KristenAndSara	-9.5%	9.2%	-9.3%	**-9.9%**
Average	-	-7.0%	-6.6%	-6.8%	**-7.41%**

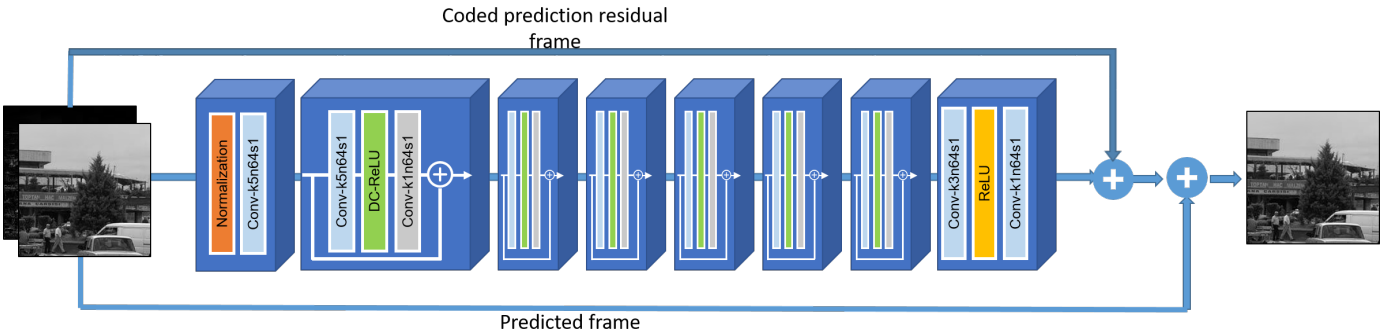

Fig. 2. Residual-based Video Restoration Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer. The predicted frame P and the coded prediction residual frame $\tilde{\mathcal{R}}$ are used as the input to Residual-VRN

C. Effect of Activation Function

In Table II, we compare the Double-Channel ReLU with popular activation functions, including ReLU, parameter ReLU, sigmoid, tanh, in the aspect of model size and performance. In the proposed models, the convolutional layers activated by DCReLU are followed by a 1×1 convolutional layer in the residual blocks. For other activation schemes, we use more activation function to keep the channels equal. We find that the DCReLU activation achieves the best result. Although parameter ReLU and tanh also has activated the negative phase, their accuracy is still inferior to the DCReLU.

TABLE II
PSNR ON JCTVC SEQUENCES ENHANCED BY DIFFERENT ACTIVATION AT QP22

QP22	Sequence	Sigmoid	Tanh	ReLU	DC-ReLU
Class B	Kimono	42.98	42.90	42.93	**43.06**
	ParkScene	41.66	41.60	41.61	**41.72**
	Cactus	40.59	40.55	40.50	**40.61**
	BasketballDrive	41.10	41.06	41.01	**41.13**
	BQTerrace	42.70	42.62	42.47	**42.73**
Class C	BasketballDrill	41.84	41.82	41.88	**41.92**
	BQMall	41.74	41.69	41.70	**41.80**
	PartyScene	41.20	41.18	41.14	**41.24**
	RaceHorses	42.23	42.19	42.18	**42.29**
Class D	BasketballPass	42.73	42.70	42.75	**42.82**
	BQSquare	41.65	41.58	41.67	**41.73**
	BlowingBubbles	41.24	41.21	41.23	**41.29**
	RaceHorses	42.41	42.37	42.48	**42.53**
Class E	FourPeople	43.94	43.84	43.91	**44.06**
	Johnny	44.08	43.99	43.97	**44.15**
	KristenAndSara	44.50	44.39	44.40	**44.61**

D. Comparing Residual-VRN with DCAD

In order to compare our Residual-based VRN with the most recently start-of-the-art method, DCAD, we train our Residual-VRN with less parameters than DCAD. Furthermore, we use the number of floating-point operations (FLOPs) to compare complexity of them. As Table III shows, Residual-based VRN achieve better performance with less parameters and less FLOPs.

TABLE III
BD-RATE RESULTS OF RESIDUAL-VRN AND DCAD.

Method		DCAD			Residual-VRN		
Parameters(k)		386.6			307.2		
FLOPs(M)		361			55.8		
Sequences		Y	U	V	Y	U	V
Class B	Kimono	-3.7%	-2.6%	-2.5%	**-9.9%**	**-6.7%**	**-7.1%**
	ParkScene	-4.6%	-3.9%	-2.9%	-1.1%	1.0%	1.8%
	Cactus	-4.3%	-4.5%	**-9.1%**	**-6.4%**	**-5.4%**	**-9.1%**
	BasketballDrive	-2.8%	-7.7%	-6.1%	**-7.4%**	**-9.6%**	**-12.8%**
	BQTerrace	-1.8%	-4.5%	**-5.0%**	**-3.1%**	**-5.8%**	-3.6%
Class C	BasketballDrill	-7.8%	-11.2%	-14.4%	**-11.5%**	**-13.2%**	**-18.7%**
	BQMall	-4.8%	**-5.7%**	-6.0%	**-7.5%**	-5.2%	**-8.1%**
	PartyScene	-2.3%	-4.5%	-5.3%	**-5.6%**	**-5.8%**	**-6.6%**
	RaceHorses	**-3.6%**	**-7.2%**	**-11.8%**	-2.8%	-4.3%	-11.6%
Class D	BasketballPass	-5.0%	-7.1%	-9.8%	**-9.4%**	**-10.2%**	**-13.9%**
	BQSquare	-3.3%	-3.2%	-6.2%	**-7.6%**	**-4.0%**	**-8.5%**
	BlowingBubbles	-4.2%	-8.7%	-8.7%	**-7.2%**	**-11.5%**	**-11.1%**
	RaceHorses	-8.4%	-10.7%	-14.3%	**-11.0%**	**-16.6%**	**-22.2%**
Class E	FourPeople	-8.3%	-6.6%	-7.2%	**-9.7%**	**-7.7%**	**-8.4%**
	Johnny	-7.3%	-9.0%	-8.1%	**-9.9%**	**-13.5%**	**-13.1%**
	KristenAndSara	-7.7%	-7.5%	-8.1%	**-9.8%**	**-10.4%**	**-12.3%**
Average	-	-5.0%	-6.5%	-8.1%	**-7.41%**	**-8.0%**	**-10.3%**

E. Comparing potential of Residual-VRN and Improved-VRN

We compare the performance of models with different number of residual blocks. As Fig. 4 shows, with the increase of the number of network layers, the performance of Residual-VRN is getting better and better. However, with the increase of the number of network layers, the performance of Improved-VRN is getting worse. Therefore, Residual-VRN holds greater potentials and is more suitable for video restoration.

F. Comparisons with State-of-the-art Approaches

We compare the performance of our networks with HEVC baseline and two state-of-the-art approaches: VRCNN and DCAD in Table IV. The subjective comparisons are showed by Fig. 3. Residual-VRN is not stable on high-resolution sequences as like sequences in ClassA and ClassB, probably because most of images in MS-COCO is low resolution. This is also a direction for us to improve in the future.

Fig. 3. Subjective comparison

TABLE IV
COMPARISONS WITH START-OF-THE-ART METHODS

	Method	VRCNN	DCAD	Improved-VRN	Residual-VRN	DR-VRN-16	DR-VRN-32	DR-VRN-64
Class B	Kimono	-2.5%	-3.7%	-7.3%	-9.9%	-10.5%	-11.5%	**-16.0%**
	ParkScene	-4.4%	-4.6%	0.7%	-1.1%	-1.8%	-2.5%	**-5.5%**
	Cactus	-4.6%	-4.3%	-5.0%	-6.4%	-6.9%	-7.7%	**-10.4%**
	BasketballDrive	-2.5%	-2.8%	-4.4%	-6.1%	-6.5%	-8.0%	**-13.8%**
	BQTerrace	-2.6%	-1.8%	-1.9%	-3.1%	-3.5%	**-4.1%**	-3.7%
Class C	BasketballDrill	-6.9%	-7.8%	-9.2%	-11.5%	-12.6%	-14.3%	**-16.8%**
	BQMall	-5.1%	-4.8%	-6.0%	-7.5%	-8.2%	-9.3%	**-11.0%**
	PartyScene	-3.6%	-2.3%	-4.8%	-5.6%	-6.0%	**-6.6%**	-6.0%
	RaceHorses	-4.2%	-3.6%	-2.1%	-2.8%	-3.0%	-3.3%	**-4.9%**
Class D	BasketballPass	-5.3%	-5.0%	-7.6%	-9.4%	-10.3%	-11.6%	**-11.8%**
	BQSquare	-3.8%	-3.3%	-6.2%	-7.6%	-8.0%	-8.9%	**-10.2%**
	BlowingBubble	-4.9%	-4.2%	-6.0%	-7.2%	-7.7%	-8.5%	**-8.7%**
	RaceHorses	-7.6%	-8.4%	-10.0%	-11.0%	-11.4%	-11.9%	**-13.1%**
Class E	FourPeople	-7.0%	-8.3%	-7.4%	-9.7%	-10.7%	-11.5%	**-14.9%**
	Johnny	-5.9%	-7.3%	-7.2%	-9.8%	-10.9%	-11.9%	**-15.0%**
	KristenAndSara	-6.7%	-7.7%	-7.6%	-9.9%	-10.7%	-11.9%	**-15.4%**
Average	-	-4.9%	-5.0%	-5.7%	-7.41%	-8.0%	-9.0%	**-11.1%**

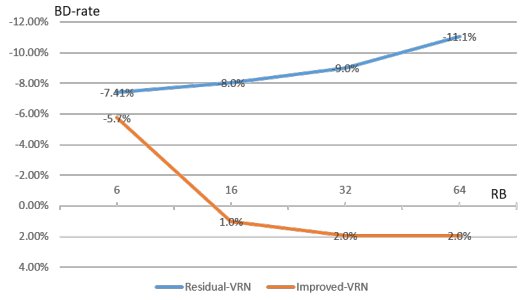

Fig. 4. BD-rate reduction varies with the number of residual blocks in the model. The horizontal axis represents the number of residual blocks, vertical axis represents BD-rate reduction

VI. CONCLUSIONS

In this paper, we proposed a novel method to improve the coding efficiency without changing the encoding algorithms of HEVC. Based on theoretical analysis, our residual-based restoration method is more efficient than the traditional way to learn a non-linear mapping from the decoded frame to an artifact-free reconstruction. Experimental results demonstrate our Residual-VRN can further improve the coding efficiency. Furthermore, we exploit the potential of Residual-VRN. Experimental result shows that Residual-VRN have greater potential than Improved-VRN.

It should be pointed that our Residual-VRN combine the HEVC algorithms and deep convolution neural network more closely than the conventional VRNs. It is worth exploring how to utilizing the information from decoding end such as PU'size and CU's distribution. Then, to design more reasonable network with the ability to make full use of the information, which will be one of our future works. On the other hand, we will extend Residual-VRN for HEVC inter coding, i.e. processing P and B frames, but there will be some necessary methods to process the MV signal and residual signal.

ACKNOWLEDGMENT

This work is partially supported by grants from the National Basic Research Program of China under grant 2015CB351806, the National Natural Science Foundation of China under contract No. U1611461, No. 61390515, and No. 61425025, also supported by grants from NVIDIA and the NVIDIA DGX-1 AI Supercomputer.

REFERENCES

[1] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[2] Gary J Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[3] Yuanying Dai, Dong Liu, and Feng Wu. A convolutional neural network approach for post-processing in hevc intra coding. In *International Conference on Multimedia Modeling*, pages 28–39. Springer, 2017.

[4] Tingting Wang, Mingjin Chen, and Hongyang Chao. A novel deep learning-based method of improving coding efficiency from the decoder-end for hevc. In *Data Compression Conference (DCC), 2017*, pages 410–419. IEEE, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Jani Lainema, Frank Bossen, Woo-Jin Han, Junghye Min, and Kemal Ugur. Intra coding of the hevc standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1792–1801, 2012.

[7] Andrey Norkin, Gisle Bjontegaard, Arild Fuldseth, Matthias Narroschke, Masaru Ikeda, Kenneth Andersson, Minhua Zhou, and Geert Van der Auwera. Hevc deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1746–1754, 2012.

[8] Chih-Ming Fu, Elena Alshina, Alexander Alshin, Yu-Wen Huang, Ching-Yeh Chen, Chia-Yang Tsai, Chih-Wei Hsu, Shaw-Min Lei, Jeong-Hoon Park, and Woo-Jin Han. Sample adaptive offset in the hevc standard. *IEEE Transactions on Circuits and Systems for Video technology*, 22(12):1755–1764, 2012.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014.

[12] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[13] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[14] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349, 2012.

[15] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 576–584, 2015.

[16] Zhangyang Wang, Ding Liu, Shiyu Chang, Qing Ling, Yingzhen Yang, and Thomas S Huang. D3: Deep dual-domain based fast restoration of jpeg-compressed images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2764–2772, 2016.

[17] Woon-Sung Park and Munchurl Kim. Cnn-based in-loop filtering for coding efficiency improvement. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*, pages 1–5. IEEE, 2016.

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

[19] Hongyang Li, Wanli Ouyang, and Xiaogang Wang. Multi-bias non-linear activation in deep neural networks. In *International conference on machine learning*, pages 221–229, 2016.

[20] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[22] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.