Residual-Based Video Restoration for HEVC Intra Coding

Li Ma1,2, Yonghong Tian#2, Tiejun Huang2

1Center for Data Science, AAIS
2National 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 $\ell_2$ 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
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 \( \xi_{msy} \), 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 \( \xi \) 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:

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

where the \( O \) is the original frame; the \( P \) represents the prediction frame; the \( \tilde{R} \) 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: \( C = D + \lambda \cdot R \), where the \( D \) and \( R \) are the distortion and bitrate for the sequence; and \( \lambda \) is the Lagrange multiplier. In frame level, if the other encoding parameters are fixed, we have

\[
m_{opt} = \arg\min_{m,\text{param}} C
\]

\[
= \arg\min_{m,\text{param}=\text{param}^*} (D_F + \lambda R(F))
\]

\[
= \arg\min_{m,\text{param}=\text{param}^*} (D(R) + \lambda R(R + P + \text{param}))
\]

\[
= \arg\min_{m,\text{param}=\text{param}^*} (D(T^{-1}Q^+QT(R)) + \lambda R(T^{-1}Q^+QT(R) + P + \text{param}))
\]

where the \( m_{opt} \) is the most efficient intra-prediction modes, and \( m_{opt} \in \{0, 1, 2, ..., 34\}^q \), where \( q = \left\lceil \frac{h}{\text{minsize}_TU} \right\rceil \times \left\lceil \frac{w}{\text{minsize}_TV} \right\rceil \), \( h, w \) denote the height and width of sequence. \( \lambda \) denote the smallest size of the prediction block. The \( \text{param} \) and \( \text{param}^* \) are the other encoding parameters and the latter is fixed. The \( D_F \) and \( D(R) \) are the distortion of the frame and the prediction residual frame. The \( D(R) \) is equal to the \( D_F \), because the distortion comes from the quantization of the residual signal’s transform coefficients. The \( \mathbb{R}(X) \) 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:

\[
\mathbb{R} = \mathbb{R}_{m_{opt}} = G(P, R, \text{para}) \tag{3}
\]

\[
(P, \mathbb{R}) = H(O, \text{para}) \tag{4}
\]

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 JPEG 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 bitrate 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
We treat it as:

\[ \text{A. Structure of VRN} \]

prediction residual frame. VRNs versatilely, and then introduce why we use coded high-level structure information learning. conventional VRNs, which is redundant and inefficient for work of them is global residual learning. We find that there utilized residual learning technique, residual learning frame-that the activation with multi-threshold performs better than threshold activation function, but Li et al. [19] have proven the positive phase. On the other hand, ReLU is a single-thought the negative phase should be considered equally with maps, and leads to the loss of some improtant details. We however, ReLU still has some limitations. Precisely, ReLU allows a network to obtain sparse representations easily. All utilize ReLU as activation function, which extensively networks.

\[ \text{B. Why Using the Coded Prediction Residual Frame} \]

The conventional VRNs treat this processing as:

\[ \text{R} = P + \tilde{R} \approx P + \mathcal{R} = O \]

The conventional VRNs treat this processing as:

\[ \hat{O} = F(R) = F(P + \tilde{R}) \approx O \]

We treat it as:

\[ \hat{O} = P + G^+(P, \tilde{R}, \text{para}) \approx O \]

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

\[ G^+(P, \tilde{R}, \text{para}) = \{ E(O - P) | H(O, \text{para}) = (P, \tilde{R}) \} \]

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

\[ O \approx P + G^{-1}(P, \tilde{R}, \text{para}) \]

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.

\[ \text{IV. Method} \]

\[ \text{A. Double-Channel ReLU Activation Function} \]

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-Channel ReLU (DCReLU) function as our activation function which is defined as

\[ DCReLU(x) = \max(x - \eta_1, 0), \min(\beta \times x - \eta_2, 0) \]

where \( \beta \) is a trainable scale parameter initialized with the value of 0.5, \( \eta_1 \) and \( \eta_2 \) are the bias thresholds that are also trainable.

\[ \text{B. Residual Block} \]

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:

\[ RB(x) = x + U(x) = x + F_2(\sigma_{DC-ReLU}(F_1(x))) \]

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 \( \text{conv} - 3 \times 3 \times 64 \) and \( \text{conv} - 1 \times 1 \times 64 \), \( U(x) \) is the residual to be learned.
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:

$$L_{\text{mix}} = \alpha \cdot G_o \cdot L_1(O, \tilde{O}) + (1 - \alpha) \cdot L_{MS-SSIM}(O, \tilde{O}) \quad (13)$$

However, their $L_{\text{mix}}$ doesn’t work on our task. We use an mixed loss function with $L_2$ loss:

$$L_{\text{mix}} = \gamma \cdot (G_o \cdot L_1(O, \tilde{O}) + L_{MS-SSIM}(O, \tilde{O})) + \sqrt{G_e \cdot L_2(O, \tilde{O})} \quad (14)$$

, 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 $\mathcal{R}$ as the input to Residual-VRN. A normalization layer is applied to reduce the difference of distribution of $P$ and $\mathcal{R}$. After a stock of convolutional layers and residual blocks, the $\mathcal{R}$ 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:

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

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 $\text{Norm}(\cdot)$ 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).

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 of our model. Finally, we compare our models with several state-of-the-art methods. For simplicity, in some comparative experiments only the luminance channel is considered for test.
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 $1 \times 1$ 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>
<thead>
<tr>
<th>Sequence</th>
<th>Sigmoid</th>
<th>Tanh</th>
<th>ReLU</th>
<th>DC-ReLU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kimono</td>
<td>42.98</td>
<td>42.90</td>
<td>42.93</td>
<td>43.06</td>
</tr>
<tr>
<td>ParkScene</td>
<td>41.66</td>
<td>41.60</td>
<td>41.61</td>
<td>41.72</td>
</tr>
<tr>
<td>Cactus</td>
<td>40.59</td>
<td>40.55</td>
<td>40.50</td>
<td>40.61</td>
</tr>
<tr>
<td>BasketballDrive</td>
<td>41.10</td>
<td>41.06</td>
<td>41.01</td>
<td>41.13</td>
</tr>
<tr>
<td>BTTerrace</td>
<td>42.70</td>
<td>42.62</td>
<td>42.47</td>
<td>42.73</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Method</th>
<th>DCAD</th>
<th>Residual-VRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLOPs(M)</td>
<td>361</td>
<td>55.8</td>
</tr>
<tr>
<td>Sequences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kimono</td>
<td>-3.7%</td>
<td>-9.9%</td>
</tr>
<tr>
<td>ParkScene</td>
<td>-4.6%</td>
<td>-6.7%</td>
</tr>
<tr>
<td>Cactus</td>
<td>-4.3%</td>
<td>-7.1%</td>
</tr>
<tr>
<td>BasketballDrive</td>
<td>-2.8%</td>
<td>-7.4%</td>
</tr>
<tr>
<td>BTTerrace</td>
<td>-1.8%</td>
<td>-3.1%</td>
</tr>
<tr>
<td>BasketballDRill</td>
<td>-7.8%</td>
<td>-11.5%</td>
</tr>
<tr>
<td>BQMall</td>
<td>-4.8%</td>
<td>-7.5%</td>
</tr>
<tr>
<td>PartyScene</td>
<td>-2.3%</td>
<td>-5.6%</td>
</tr>
<tr>
<td>RaceHorses</td>
<td>-3.6%</td>
<td>-2.9%</td>
</tr>
<tr>
<td>BasketballPass</td>
<td>-5.0%</td>
<td>-9.4%</td>
</tr>
<tr>
<td>BQMall</td>
<td>-4.8%</td>
<td>-5.7%</td>
</tr>
<tr>
<td>PartyScene</td>
<td>-2.3%</td>
<td>-5.9%</td>
</tr>
<tr>
<td>RaceHorses</td>
<td>-3.3%</td>
<td>-5.8%</td>
</tr>
<tr>
<td>Average</td>
<td>-5.0%</td>
<td>-6.4%</td>
</tr>
</tbody>
</table>

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.
TABLE IV

COMPARISONS WITH START-OF-THE-ART METHODS

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

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 utilize the information from decoding end such as PU’s 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


