DECODER-SIDE HEVC QUALITY ENHANCEMENT WITH SCALABLE CONVOLUTIONAL NEURAL NETWORK

Ren Yang[†], *Mai Xu*[†*] *and Zulin Wang*[†‡]

[†]School of Electronic and Information Engineering, Beihang University, China
[‡]Collaborative Innovation Center of Geospatial Technology, Wuhan, China
{yangren, maixu, wzulin}@buaa.edu.cn

ABSTRACT

The latest High Efficiency Video Coding (HEVC) has been increasingly used to generate video streams over Internet. However, the decoded HEVC video streams may incur severe quality degradation, especially at low bit-rates. Thus, it is necessary to enhance visual quality of HEVC videos at the decoder side. To this end, we propose in this paper a Decoder-side Scalable Convolutional Neural Network (DS-CNN) approach to achieve quality enhancement for HEVC, which does not require any modification of the encoder. In particular, our DS-CNN approach learns a model of Convolutional Neural Network (CNN) to reduce distortion of both I and B/P frames in HEVC. It is different from the existing CNN-based quality enhancement approaches, which only handle intra coding distortion, thus not suitable for B/P frames. Furthermore, a scalable structure is included in our DS-CNN, such that the computational complexity of our DS-CNN approach is adjustable to the changing computational resources. Finally, the experimental results show the effectiveness of our DS-CNN approach in enhancing quality for both I and B/P frames of HEVC.

Index Terms— HEVC, quality improvement, convolutional neural network

1. INTRODUCTION

High Efficiency Video Coding (HEVC) [1] is the state-of-the-art video coding standard, which is able to reduce the bit-rate of H.264/AVC to around 60% with similar subjective quality [2]. Thanks to its outstanding coding efficiency, HEVC has been increasingly applied to generate video streams in recent multimedia applications. However, like former video coding standards, HEVC videos also incur artifacts, such as blocking artifacts, ringing effects, blurring, etc., especially at low bit-rates. Sometimes, such artifacts may cause severe degradation on Quality of Experience (QoE) at the decoder side. Therefore, it is necessary to study on enhancing visual quality of HEVC videos at the decoder side.

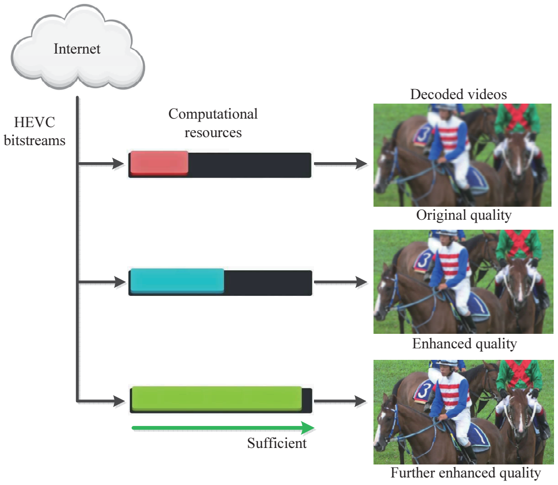


Fig. 1. An example for application scenario of our DS-CNN approach. As shown, our approach makes a trade-off between quality enhancement and computational complexity. When computational resources are not enough, our DS-CNN achieves some quality improvement of decoded HEVC videos. Once computational resources are sufficient, the quality of decoded videos can be further enhanced.

The past decade has witnessed the growing interests on quality enhancement of decoded images or videos. However, most of the existing works [3, 4, 5, 6, 7, 8] focus on enhancing visual quality of decoded images. For example, the method proposed by Liew *et al.* [3] reduces blocking artifacts of block-coded images using overcomplete wavelet representation. Foi *et al.* [4] applied pointwise Shape-Adaptive DCT (SA-DCT) to reduce blocking and ringing effects caused by JPEG compression. Later, Wang *et al.* [5] proposed to filter the boundaries between blocks for reducing blocky artifacts of JPEG images. Recently, Jancsary *et al.* [6] achieved JPEG image deblocking by taking advantage of Regression Tree Fields (RTF). Moreover, there exist some sparse coding methods for removing JPEG artifacts, such as [7] and [8].

In recent years, Convolutional Neural Network (CNN) [9], as a kind of deep learning approach, has made impressive achievements in computer vision and image processing tasks [10, 11, 12, 13]. Most recently, CNN has also been applied to improve visual quality of decoded images. Dong *et al.* [14] designed a four-layer CNN, named AR-CNN, for improving

[*]Corresponding author: Mai Xu (maixu@buaa.edu.cn). This work was supported by the NSFC projects under Grants 61573037, 61202139, and 61471022, and Fok Ying-Tong education foundation under grant 151061.

978-1-5090-6067-2/17/$31.00 ©2017 IEEE

the quality of JPEG images. Wang *et al.* [15] investigated another deep network, called $\mathbf{D}^3$, for JPEG image restoration. As reported in the experiments of [14, 15], these CNN-based approaches outperform other conventional methods, such as [4, 6, 7]. The outstanding performance of [14, 15] illuminates the promise of CNN approach in quality enhancement for images or videos.

To improve the quality of decoded videos, many works [16, 17, 18] were proposed for the latest HEVC standard. Han *et al.* [16] developed a high performance in-loop filter for HEVC, which is added after the original in-loop filter in HEVC encoder and decoder. The bit-rate reduces averagely about 2% by using this approach. Park and Kim [17] designed a CNN to replace the Sample Adaptive Offset (SAO) filter in HEVC encoder and decoder. Later, based on AR-CNN [14], Dai *et al.* [18] proposed a CNN (named VRCNN) to replace the in-loop filters in HEVC intra coding, which successfully makes 4.6% bit-rate reduction. It has been verified that VRCNN [18] performs better than [16], further showing the greater ability of CNN in video quality enhancement. However, all above methods require the modification of HEVC encoder, and thus they are unpractical for enhancing the quality of videos that have been already encoded. In addition, despite benefitting from AR-CNN [14], the latest CNN-based quality enhancement work [18] for HEVC can only handle the intra mode coding. In other words, the distortion of inter frames (i.e., B and P frames) is not taken into consideration in [18]. As such, [18] is not well suitable for B/P frames in HEVC, whose number is usually much more than I frames. Furthermore, as far as we know, there exists no work for achieving computation-scalable quality enhancement for decoded images or videos, which can adapt to the varying computational resources.

To overcome the above disadvantages, we propose in this paper a Decoder-side Scalable CNN (DS-CNN) for enhancing the visual quality of decoded HEVC videos. The proposed DS-CNN has the ability to extract features of both HEVC intra and inter coding. As a result, DS-CNN is suitable for enhancing the quality of both I and B/P frames. Moreover, a scalable structure is included in our DS-CNN, to meet the requirement of variable computational resource conditions. Fig. 1 shows a possible application scenario of our approach[1]. It is worth pointing out that the HEVC encoder does not need to be modified, when applying our approach at the decoder side. The main contributions of our approach are two-fold:

1) We propose the DS-CNN model to reduce the distortion of both I and B/P frames, thus achieving HEVC quality enhancement at the decoder side;

2) We design a scalable structure with two sub-networks in our DS-CNN, such that the quality enhancement of decoded HEVC videos can be adjustable to varying computational resources.

2. OVERVIEW OF AR-CNN

In [14], AR-CNN is designed to improve the visual quality of encoded JPEG images. To our best knowledge, it is the first work to apply CNN in improving visual quality of encoded images, which has shown great success in quality enhancement. Therefore, it can be seen as the foundation of our approach. In the following, we briefly review the overall architecture of AR-CNN.

Table 1. Configuration of AR-CNN [14]

Layer index	Conv 1	Conv 2	Conv 3	Conv 4
Filter size	9×9	7×7	1×1	5×5
Filter number	64	32	16	1

In AR-CNN, there are four convolutional layers without any pooling or fully-connected layer. Specifically, the four layers of AR-CNN play the roles of feature extraction, feature denoising, non-linear mapping and reconstruction. The input image is denoted by $\mathbf{Y}$, and the output of the i-th convolutional layer is defined as $F_i(\mathbf{Y})$. Then, the AR-CNN network can be expressed as

$$
\begin{aligned}
F_0(\mathbf{Y}) &= \mathbf{Y}, && (1)\\
F_i(\mathbf{Y}) &= \max(0, W_i * F_{i-1}(\mathbf{Y}) + B_i),\ i \in \{1,2,3\}, && (2)\\
F_4(\mathbf{Y}) &= W_4 * F_3(\mathbf{Y}) + B_4, && (3)
\end{aligned}
$$

where W_i and B_i are the weights and bias matrices of the i-th layer, and $*$ indicates the convolution operator. Note that $\max(0, x)$, known as Rectified Linear Unit (ReLU), is adopted in the first three layers as the non-linear activation function. The configuration of AR-CNN is summarized in Table 1.

3. ARCHITECTURE OF DS-CNN

In this section, we concentrate on the architecture of our DS-CNN. DS-CNN includes two sub-networks, i.e., DS-CNN-I and DS-CNN-B, designed for quality enhancement of I and B/P frames, respectively. The detailed architecture is to be discussed in the following.

3.1. DS-CNN-I

Training and validation sets. First of all, we select the training and validation sets for DS-CNN-I, to tune its architecture and parameters. Here, our training and validation sets are the same as AR-CNN [14], which are selected from BSDS500 database [19]. Because DS-CNN-I aims at reducing distortion of I frames in HEVC, we encode all training images with HEVC All Intra (AI) mode. For training, we decompose the ground-truth and HEVC coded images into image patches with the size of 40×40, using the stride of 10. As such, the training set with 400 images provides totally 522,000 pairs of training samples. Similarly, 34,500 pairs of validation samples are obtained.

Loss function. We apply Mean Squared Error (MSE) as the loss function of our DS-CNN-I. Let $\{\mathbf{X}_n\}_{n=1}^{N}$ be the set of raw image patches, seen as ground-truth, and $\{\mathbf{Y}_n\}_{n=1}^{N}$

[1] Note that Fig. 1 only illustrates an example application, and it is not the real experimental results. The experimental results are shown in Section 4.

be patches of their corresponding compressed images. Here, $\{\mathbf{Y}_n\}_{n=1}^N$ are input samples, whereas $\{\mathbf{X}_n\}_{n=1}^N$ are the corresponding target output. Define $F(\cdot)$ as the output of DS-CNN-I. Then, the loss function is as follows,

$$L(\Theta) \quad = \quad \frac{1}{N}\sum_{n=1}^{N}||F(\mathbf{Y}_n;\Theta)-\mathbf{X}_n||_2^2, \qquad (4)$$

where $\Theta = \{W_i, B_i\}$ stands for the weights and bias in DS-CNN-I. This loss function is minimized by stochastic gradient decent algorithm with the standard Back-Propagation (BP). The batch size for training DS-CNN-I is set as 128.

Architecture. The architecture of DS-CNN-I is designed according to the following observations.

Observation 1. The distortion caused by HEVC intra coding is with more features than JPEG.

Proof. We prove this observation from both theoretical and experimental analysis. Theoretically, HEVC intra coding is more complicated comparing to JPEG. For example, HEVC supports different sizes of Discrete Cosine Transform (DCT), including 4×4, 8×8, 16×16 and 32×32 [1], while JPEG only adopts 8×8 DCT [20]. Moreover, HEVC intra-picture prediction has 33 different directional orientations [1], much more complex than the intra prediction in JPEG [20].

Next, we further prove Observation 1 from experimental perspective, by testing CNN on the validation set at different configuration. In CNN, the convolutional filters are used to extract features from the input images [9] for quality enhancement. As such, more convolutional filters should be used to handle the input images with more distortion-related features. Hence, based on AR-CNN, we design AR-CNN-1 with larger number of filters. Then, we compare the performance of AR-CNN-1 with AR-CNN at QP = 42 on the validation set. The configuration of AR-CNN-1 are shown in Table 2, and its performance compared with AR-CNN is shown in Fig. 2. It can be seen from Table 2 that CNN with more filters performs better on enhancing the quality of HEVC I frames. Therefore, Observation 1 can be proved.

Table 2. Configuration and performance of AR-CNN and AR-CNN-1/2.

	AR-CNN	AR-CNN-1	AR-CNN-2
Filter size	9-7-1-5	9-7-1-5	9-7-3-1-5
Filter number	64-32-16-1	128-64-32-1	128-64-64-32-1

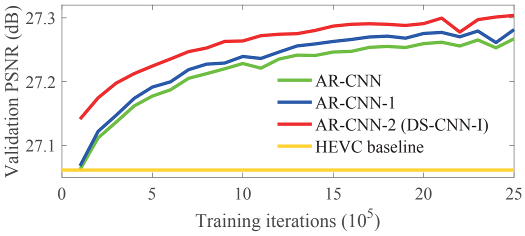


Fig. 2. Performance of AR-CNN and AR-CNN-1/2.

Observation 2. AR-CNN-1 with one more convolutional layer can extract more effective distortion-related features for HEVC, thus leading to better performance.

Proof. Recall that AR-CNN is a 4-layer network, in which Conv 1 is used to extract features and Conv 2 is designed for feature denoising [14]. It has been proved in [14] that AR-CNN successfully improves the performance comparing to the 3-layer CNN without feature denoising layer, and transferring AR-CNN to 5 layers can achieve better performance on JPEG restoration. Motivated by this, for HEVC, we also extend AR-CNN-1 to AR-CNN-2, which includes one more layer after Conv 2 to further denoise the features. The configuration of AR-CNN-2 is shown in Table 2. As such, AR-CNN-2 has 5 convolutional layers.[2]

Then, we test AR-CNN-2 on the validation set at QP = 42. As shown in Fig. 2, AR-CNN-2 outperforms AR-CNN-1 for HEVC quality enhancement of I frames. Therefore, it can be validated that Conv 3 succeeds in further denoising the feature maps for HEVC encoded images. Finally, Observation 2 is proved.

According to Observations 1 and 2, we use AR-CNN-2 as our DS-CNN-I to enhance quality of I frames of decoded HEVC videos. The architecture of DS-CNN-I is shown in Fig. 3, and its configuration is shown in Table 3. The formulation of DS-CNN-I can be expressed as

$$F_0(\mathbf{Y}) = \mathbf{Y}, \qquad (5)$$

$$F_i(\mathbf{Y}) = \max(0, W_i * F_{i-1}(\mathbf{Y}) + B_i),\ i \in \{1,2,3,4\}, (6)$$

$$F_5(\mathbf{Y}) = W_5 * F_4(\mathbf{Y}) + B_5. \qquad (7)$$

Note that ReLU, i.e., $\max(0, x)$, is adopted in the layers of Conv 1-4 as the non-linear activation function.

Table 3. Configuration of DS-CNN.

DS-CNN-I DS-CNN-B	Conv 1 Conv 6	Conv 2 Conv 7	Conv 3 Conv 8	Conv 4 Conv 9	Conv 5 Conv 10
Filter size	9×9	7×7	3×3	1×1	5×5
Filter number	128	64	64	32	1
W learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-5}
B learning rate	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}

3.2. DS-CNN-B

Training and validation sets. The training set for DS-CNN-B includes 26 sequences randomly selected from JCT-VC database [21] and Xiph.org [22]. Moreover, *RaceHorses* (480p) and *BasketballPass* (240p) of JCT-VC database are chosen as validation sequences.

Since DS-CNN-B is designed for quality enhancement of B/P frames, the training sequences are all encoded by HEVC with Random Access (RA) configurations. We randomly select 10 B frames from each training sequence, and decompose the ground-truth and encoded frames into pairs of 40×40 image patches, with the stride being 15. This way, we obtain 963,500 training sample pairs. Similarly, 43,980 validation sample pairs are obtained. The loss function is the same as (4) in Section 3.1. The batch size is also set to be 128 when

[2] According to [14], 5-layer networks are sensitive to initialization and hard to convergence. Hence, to ensure the convergence of AR-CNN-2, Conv 1/2 in AR-CNN-2 are fine-tuned from Conv 1/2 of the pretrained AR-CNN-1.

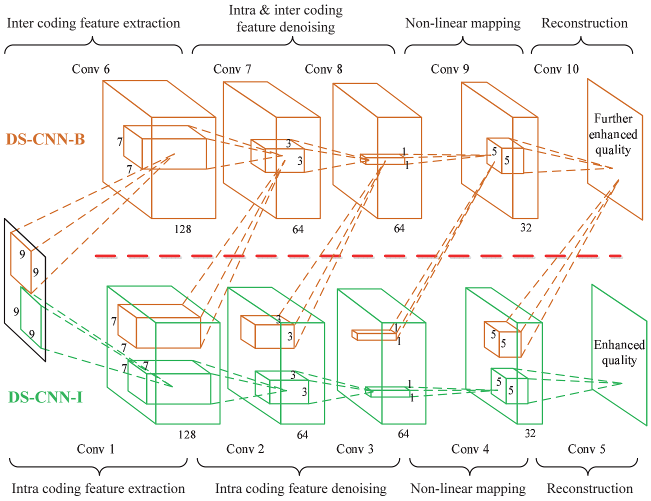


Fig. 3. Architecture of DS-CNN.

applying the stochastic gradient decent algorithm to train DS-CNN-B.

Architecture. In HEVC coding, only intra-picture coding is used in I frames, while both intra- and inter-picture coding is applied on B/P frames [1]. As such, we make use of the features of both HEVC intra and inter coding, to achieve the quality improvement of B/P frames. Accordingly, we design DS-CNN-B also containing 5 convolutional layers, as shown in Fig 3. In DS-CNN-B, the layer of Conv 6 is used to extract distortion-related features of HEVC inter coding from the input frames. Recall that Conv 1 is to extract intra coding features. Then, the outputs of Conv 1 and Conv 6 are concatenated, and are both convolved by Conv 7. Thus, Conv 7 denoises the features of both intra and inter coding. Conv 8-10 in DS-CNN-B are designed in the similar way. Finally, the layer of Conv 10 in DS-CNN-B can reconstruct the B/P frames, based on distortion-related features of both HEVC intra and inter coding, thus achieving quality enhancement on B/P frames.

In Conv 7-9, we define $W_i^{(1)}$ and $W_i^{(2)}$ as the weights of Conv i used to convolve the data in DS-CNN-I and DS-CNN-B, respectively. The formulation of DS-CNN-B can be expressed as

$$F_6(\mathbf{Y}) = \max(0, W_6 * F_0(\mathbf{Y}) + B_5), \tag{8}$$

$$F_i(\mathbf{Y}) = \max(0, W_i^{(1)} * F_{i-6}(\mathbf{Y}) + W_i^{(2)} * F_{i-1}(\mathbf{Y}) + B_i), \quad i \in \{7, 8, 9\}, \tag{9}$$

$$F_{10}(\mathbf{Y}) = W_{10} * F_9(\mathbf{Y}) + B_{10}, \tag{10}$$

where ReLU, i.e., $\max(0, x)$, is adopted in Conv 6-9 as the non-linear activation function. In DS-CNN-B, we set the filter sizes and filter numbers the same as the corresponding layers in DS-CNN-I, as shown in Table 3.

Training procedure. Before training DS-CNN-B, DS-CNN-I needs to be trained following the procedure of Section 3.1. After trained by HEVC intra coded images, the convolutional layers in DS-CNN-I have the ability to handle intra coding features. Afterwards, DS-CNN-B is trained with the learning rate shown in Table 3. Note that when training DS-CNN-B, the weights and bias of DS-CNN-I keep unchanged.

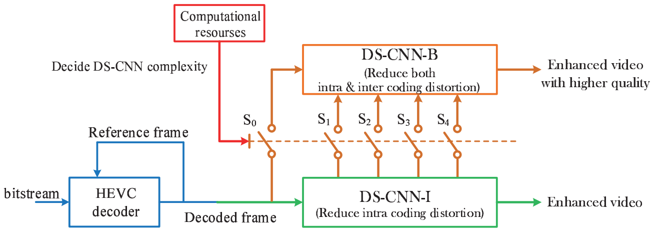


Fig. 4. Scalable structure of DS-CNN.

3.3. Scalable structure of DS-CNN

To meet the varying computational resources of different decoding devices, we propose a scalable structure in our DS-CNN. As shown in Fig. 4, we adopt switches $\{\mathrm{S}_i\}_{i=0}^4$ to control computational complexity of quality enhancement for HEVC. Note that switches $\{\mathrm{S}_i\}_{i=0}^4$ decide whether to enable the convolutional layers of DS-CNN-B.

Once the computational resources are not sufficient, the switches $\{\mathrm{S}_i\}_{i=0}^4$ are turned off, and only DS-CNN-I is in use at the decoder. Here, DS-CNN-I is applied on the whole decoded HEVC video, including both I and B/P frames. Since HEVC intra-picture coding is used in both I and B/P frames, DS-CNN-I, trained by I frames, can also achieve quality enhancement on B/P frames due to the reduction of intra coding distortion. When the computational resources are sufficient, $\{\mathrm{S}_i\}_{i=0}^4$ are turned on, and DS-CNN-B starts to work based on the output from the layers Conv 1-4 of DS-CNN-I. Because of the reduction of inter coding distortion, the quality of B/P frames can be further enhanced by DS-CNN-B, at the cost of higher computational complexity.

It is worth pointing out that there is no need to modify the HEVC encoder, when applying our DS-CNN approach at the decoder side. As Fig. 4 shows, when decoding, we still use the original decoded frames (before being enhanced by DS-CNN) as reference frames. As such, our DS-CNN does not affect HEVC decoding process, and therefore the HEVC encoder is not required to be modified. This makes our DS-CNN practical for already encoded videos.

4. EXPERIMENTS

In this section, the experimental results are presented to validate the effectiveness of our DS-CNN approach, comparing with the latest quality enhancement approaches AR-CNN [14] and VRCNN [18]. In this paper, we do not compare DS-CNN with the JPEG restoration approach $\mathbf{D}^3$ [15], because $\mathbf{D}^3$ takes advantage of the prior knowledge of JPEG compression scheme, so that it cannot to be used for HEVC. In the following, we present the settings of our experiments in Section 4.1, and evaluate the performance of quality enhancement of our DS-CNN approach in Section 4.2.

4.1. Settings

We test our approach on 9 sequences from JCT-VC database [21], non-overlapping with the training and validation sets. The test set includes *BQTerrace*, *Cactus* and *BasketballDrive*

Table 4. Quality enhancement performance of I frames and B frames.

QP	Class	Sequence	Δ PSNR (dB)						
			VRCNN	AR-CNN		AR-CNN*	DS-CNN-I		DS-CNN-B
			I frames	I frames	B frames	B frames	I frames	B frames	B frames
42	B	*BQTerrace*	0.3127	0.3180	0.2530	0.2103	**0.3789**	0.2774	**0.2987**
		Cactus	0.1754	0.1452	0.1289	0.1584	**0.2001**	0.1671	**0.2186**
		BasketballDrive	0.1776	0.1737	0.1478	0.1603	**0.2281**	0.1842	**0.2039**
	C	*BQMall*	0.2946	0.2679	0.2246	0.2199	**0.3433**	0.2725	**0.2749**
	D	*RaceHorses* (240p)	0.4117	0.3697	0.1997	0.2004	**0.4320**	0.2423	**0.2505**
	E	*FourPeople*	0.4060	0.3887	0.3488	0.3839	**0.4791**	0.4126	**0.4317**
		Johnny	0.2823	0.2837	0.2371	0.2172	**0.3363**	0.2468	**0.2799**
	E’	*Vidyo1*	0.3619	0.3064	0.2959	0.3038	**0.4086**	0.3654	**0.3933**
		Vidyo3	0.2126	0.2892	0.2758	0.3085	**0.3468**	0.3136	**0.3560**
	AVERAGE		0.2928	0.2825	0.2346	0.2403	**0.3504**	0.2758	**0.3008**
47	**AVERAGE**		0.2940	0.2934	0.2482	0.2693	**0.3413**	0.2871	**0.3051**

from Class B, *BQMall* from Class C, *BasketballPass* from Class D and *FourPeople*, *Johnny*, *Vidyo1*, *Vidyo4* from Class E/E$'$. All test sequences are encoded by HEVC RA mode, using HM 16.0 at QP = 42 and 47. The HM encoder is set by the default configuration of *encoder_randomaccess_main.cfg*.

4.2. Performance of quality enhancement

In this subsection, we evaluate the performance of quality enhancement of our DS-CNN, comparing to the conventional AR-CNN [14] and VRCNN [18]. The quality enhancement is measured by Y-PSNR improvement (Δ PSNR). The results are shown in Table 4. Recall that AR-CNN is a quality enhancement approach for JPEG images [14], and VRCNN is designed for HEVC intra mode only [18]. Therefore, AR-CNN is retrained on HEVC encoded images, and the training patches for AR-CNN and VRCNN are the same as DS-CNN-I. Furthermore, for fair comparison on the performance of B frames, we also train AR-CNN using the training patches of video sequences for DS-CNN-B, and we name it as AR-CNN* in order to distinguish.

Performance of I frames. It can be seen from Table 4, the proposed DS-CNN-I obviously outperforms AR-CNN and VRCNN on the I frames over all test sequences. At QP = 42, the averaged Δ PSNR (0.3504 dB) of our DS-CNN-I is 24% higher than AR-CNN (0.2825 dB) and 20% higher than VRCNN (0.2928 dB). In particular, DS-CNN-I achieves up to 0.4791 dB Y-PSNR improvement on I frames, while the highest improvements of AR-CNN and VRCNN are 0.3887 dB and 0.3612 dB, respectively. Similar results can be found at QP = 47. In summary, our proposed DS-CNN-I performs best in quality enhancement of HEVC I frames among three approaches.

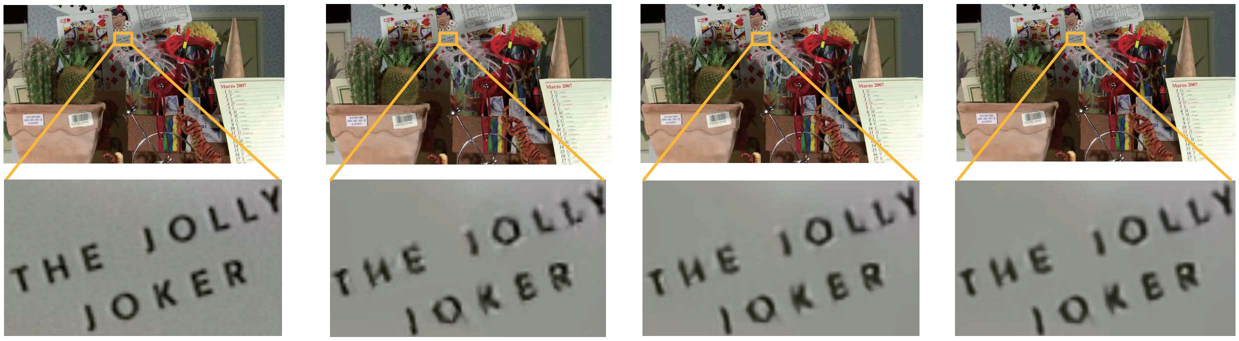
Performance of B frames. As shown in Table 4, the quality enhancement performance of our DS-CNN is much better than AR-CNN and AR-CNN* over all test sequences. Note that we do not compare with VRCNN for B frames, because VRCNN is designed to improve coding efficiency of HEVC intra mode only. If applying VRCNN to B frames, the HEVC encoder needs to be modified. Table 4 shows that our DS-CNN-I also has the ability to enhance the quality of B frames, due to the reduction of intra coding distortion. DS-CNN-I averagely makes 18% more PSNR improvement of B frames comparing to AR-CNN at QP = 42.

Furthermore, our DS-CNN-B achieves more improvement on decoded B frames than DS-CNN-I, because of its specific design for enhancing quality of B frames. At QP = 42, DS-CNN-B reaches 0.3008 dB PSNR improvement of B frames averaged over all test sequences, which is 28% and 25% higher than AR-CNN (0.2346 dB) and AR-CNN* (0.2403 dB), respectively. Similar results can be found for QP = 47. Fig. 5 shows the subjective results on a B frame of *Cactus* at QP = 42. It can be seen that our DS-CNN-B effectively reduces the artifacts caused by HEVC compression, and performs better than AR-CNN*. Hence, the effectiveness of our DS-CNN for quality enhancement of B frames can be validated.

Complexity-enhancement performance. In our DS-CNN, there is a scalable structure, designed to make a trade-off between computational complexity and quality enhancement. We evaluate the computational complexity via running time on a Ubuntu PC with Inter(R) Core(TM) i7-4790K CPU and one GeForce GTX 1080 GPU. When the switches $\{\mathrm{S}_i\}_{i=0}^4$ are turned off, only DS-CNN-I is used to enhance both I and B frames. The averaged running time is $0.719(\pm 0.039)$ ms per Coding Tree Unit (CTU). When $\{\mathrm{S}_i\}_{i=0}^4$ are turned on, and the whole DS-CNN consumes $1.94(\pm 0.100)$ ms per CTU. Here, the CTU size is set as 64×64 pixels. Note that when applying the whole DS-CNN, I frames and B frames are enhanced by DS-CNN-I and DS-CNN-B, respectively. Thus, switching on $\{\mathrm{S}_i\}_{i=0}^4$ leads to $170(\pm 6)\%$ complexity increment. Fig. 6 shows the complexity-enhancement performance of four test sequences. It can be seen from Table 4 and Fig. 6 that whole DS-CNN achieves averagely 9% and up to 30% extra PSNR improvement comparing with DS-CNN-I when QP = 42, at the cost of $\sim$1.70 times increment of computational complexity.

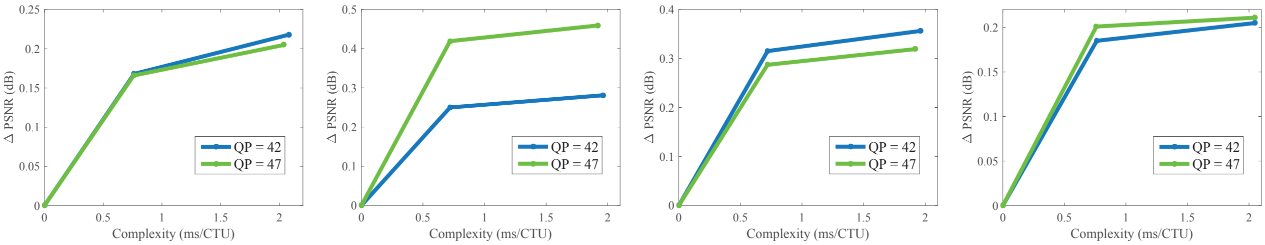
5. CONCLUSION

In this paper, we have proposed a CNN-based approach, namely DS-CNN, to enhance the quality of HEVC decode videos. Our DS-CNN does not need any modification of HEVC encoder, so that it is practical for already encoded videos. Moreover, our DS-CNN has the ability to handle both intra and inter coding distortion. Therefore, it performs well



Ground-truth HEVC baseline AR-CNN* DS-CNN-B

Fig. 5. Results on the 9-th frame (B frame) of *Cactus* at QP = 42.



(a) *Cactus* (b) *Johnny* (c) *Vidyo3* (d) *BasketballDrive*

Fig. 6. Complexity-enhancement performance of our DS-CNN.

on both HEVC I frames and B frames. We also designed a scalable structure in our DS-CNN for achieving the trade-off between quality enhancement and computational complexity, which improves the flexibility of our DS-CNN and makes it adjustable to the changing computational resources. Finally, experimental results were presented to show that our approach performs better than other state-of-the-art quality enhancement approaches.

6. REFERENCES

[1] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE TCSVT*, pp. 1649–1668, 2012.

[2] Thiow Keng Tan, Rajitha Weerakkody, Marta Mrak, Naeem Ramzan, Vittorio Baroncini, Jens-Rainer Ohm, and Gary J Sullivan, "Video quality evaluation methodology and verification testing of HEVC compression performance," *IEEE TCSVT*, pp. 76–90, 2016.

[3] AW-C Liew and Hong Yan, "Blocking artifacts suppression in block-coded images using overcomplete wavelet representation," *IEEE TCSVT*, pp. 450–461, 2004.

[4] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE TIP*, 2007.

[5] Ci Wang, Jun Zhou, and Shu Liu, "Adaptive non-local means filter for image deblocking," *Signal Processing: Image Communication*, pp. 522–530, 2013.

[6] Jeremy Jancsary, Sebastian Nowozin, and Carsten Rother, "Loss-specific training of non-parametric image restoration models: A new state of the art," in *ECCV*, 2012.

[7] Huibin Chang, Michael K Ng, and Tieyong Zeng, "Reducing artifacts in JPEG decompression via a learned dictionary," *IEEE TSP*, pp. 718–728, 2014.

[8] Cheolkon Jung, Licheng Jiao, Hongtao Qi, and Tian Sun, "Image deblocking via sparse representation," *Signal Processing: Image Communication*, pp. 663–677, 2012.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[14] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang, "Compression artifacts reduction by a deep convolutional network," in *ICCV*, 2015.

[15] Zhangyang Wang, Ding Liu, Shiyu Chang, Qing Ling, and Thomas S Huang, "D3: Deep dual-domain based fast restoration of jpeg-compressed images," *arXiv preprint arXiv:1601.04149*, 2016.

[16] Qinglong Han and Wai-Kuen Cham, "High performance loop filter for HEVC," in *ICIP*, 2015.

[17] Woon-Sung Park and Munchurl Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *IVMSP*. IEEE, 2016.

[18] Yuanying Dai, Dong Liu, and Feng Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," *arXiv preprint arXiv:1608.06690*, 2016.

[19] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE TPAMI*, pp. 898–916, 2011.

[20] Gregory K Wallace, "The JPEG still picture compression standard," *IEEE TCE*, pp. xviii–xxxiv, 1992.

[21] Frank Bossen et al., "Common test conditions and software reference configurations," *Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-F900*, 2011.

[22] Xiph.org, "Xiph.org video test media," https://media.xiph.org/video/derf/.