NEURAL NETWORK BASED INTER PREDICTION FOR HEVC

Yang Wang[1]*, Xiaopeng Fan*[1]*, Chuanmin Jia*[2]*, Debin Zhao*[1]*, and Wen Gao*[2]

[1] Harbin Institute of Technology, China, [2] Peking University, China
[1] {wangyang.cs, fxp, dbzhao}@hit.edu.cn, [2] {cmjia, wgao}@pku.edu.cn

ABSTRACT

HEVC is the latest video coding standard, in which inter prediction plays an important role to reduce the temporal redundancy. The accuracy of inter prediction is limited since only temporal information is used in conventional algorithms. In this paper, we propose a neural network based inter prediction algorithm for HEVC by using the spatial-temporal information. In the proposed algorithm, we first design a neural network architecture consisting of a fully connected network (FCN) and a convolutional neural network (CNN). Then the spatial neighboring pixels and the temporal neighboring pixels are inputted into FCN. The output of FCN and the prediction of current block are inputted into CNN, which will results the more accurate prediction of current block. Experimental results demonstrate that the proposed method can achieve average 1.7% (up to 8.6%) BD-rate reduction in low delay P test condition compared to HM 16.9.

***Index Terms*—** HEVC, Inter prediction, Neural network, Fully connected network, Convolutional neural network

1. INTRODUCTION

High efficiency video coding (HEVC) standard, developed by the Joint Collaborative Team on Video Coding (JCT-VC) [1], becomes the state-of-the-art video coding standard. It can provide a similar perceptual quality with about 50% bitrate saving compared with its predecessor H.264/AVC [2]. Inter prediction plays an important role in HEVC to achieve this remarkable improvement. In inter prediction of HEVC, the prediction of current block is obtained by straightly copying or interpolating a block from the reference picture. However, the temporal illumination variation and the correlation between current block and its neighboring pixels are not considered, which may hamper the accuracy of inter prediction.

To improve the accuracy of inter prediction, some conventional research has been presented. Yin *et al*. proposed a localized weighted prediction method to handle the local illumination variations [3]. An offset is estimated for each block by using the neighboring pixels of the block and is used to compensate the illumination variations. Jeong *et al*. proposed to replace the skip mode by a skip with offset method [4], in which the offset is estimated similarly as in [3]. Zhang *et al*. proposed to employ an improved linear regression model to improve the accuracy of inter prediction in HEVC [5]. The weighted parameters in the model are estimated by referring the reconstructed neighboring pixels and temporal reference block.

Recently, deep learning has achieved impressive results on artifacts reduction for images and videos [6, 7, 8, 9, 10, 11]. Dong *et al*. proposed a convolutional neural network for image super-resolution (SRCNN) [6]. Dong *et al*. proposed a convolutional neural network for JPEG image artifacts reduction (ARCNN) [7]. To reduce JPEG image artifacts, a deep dual-domain based restoration model was investigated in [8]. Dai *et al*. proposed a variable-filter-size convolutional neural network (VRCNN) for post-processing in HEVC intra coding [9]. Wang *et al*. proposed a very deep convolutional neural network to remove the artifacts of HEVC compressed video, in which the network consisted of 10 convolutional layers [10]. Yang *et al*. proposed a decoder-side scalable convolutional neural network approach to enhance the quality of HEVC compressed video [11].

Apart from algorithms for artifacts reduction, deep learning is also investigated to improve the compression performance of HEVC. Park *et al*. straightly integrated SRCNN in HEVC to replace the deblocking filter and sample adaptive offset (SAO) [12]. Jia *et al*. proposed a spatial-temporal residue network and integrated it in HEVC as an additional filtering method, located after SAO, in which spatial-temporal coherences were jointly exploited to infer the pristine visual signal [13]. Li *et al*. proposed a fully connected network for intra prediction, where the inputs are multiple reference lines of the current block and the output is the prediction of current block [14]. Li *et al*. proposed a CNN-based block up-sampling scheme for intra frame coding in HEVC, in which a new CNN structure was explored for up-sampling [15].

In this paper, we propose a neural network based inter prediction algorithm (NNIP) for HEVC. Specially, we propose a neural network architecture to improve the accuracy of inter prediction, in which the network consists of a fully connected network (FCN) and a convolutional neural network (CNN).

This work was supported by the Major State Basic Research Development Program of China (973 Program) under Grant 2015CB351804 and the National Science Foundation of China (NSFC) under grants 61472101, 61631017.

978-1-5386-1737-3/18/$31.00

The inputs of FCN are the spatial neighboring pixels and the temporal neighboring pixels. The inputs of CNN are the prediction of current block and the output of FCN. The enhanced prediction of current block can be obtained by using the neural network. Note that VRCNN is used as CNN in this paper. Experimental results demonstrate that when compared to HM 16.9, the proposed NNIP method can achieve about 1.7% (up to 8.6%) BD-rate reduction in low delay P (LDP) test condition.

The remainder of this paper is organized as follows. Section II gives a brief overview of inter prediction in HEVC and VRCNN. The proposed neural network based inter prediction algorithm is introduced in section III. In section IV, experimental results are shown, followed by the conclusion in section V.

2. BACKGROUND

2.1. Inter prediction in HEVC

Inter prediction plays an important role in HEVC. The compression performance can be increased by improving the accuracy of inter prediction. Coding unit (CU) is assigned a particular prediction mode, either intra prediction or inter prediction. Each CU is associated with one or more prediction units (PU). As shown in Fig. 1, there are eight partition modes for inter predicted CU.

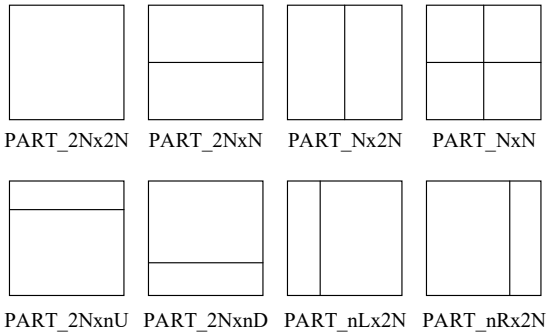

Fig. 1. Partition modes for inter PU.

The simplified diagram of encoding and decoding with inter prediction is shown in Fig. 2. In encoder, inter prediction is used to generate the prediction of current block, whether merge mode or normal inter mode is used. Motion estimation is used in normal inter mode to generate the motion vector, which is used to generate the prediction in motion compensation. Then the transform, quantization and entropy coding processes are used to generate the bitstream. In decoder, the motion vector and the residue are derived from entropy decoding process. Reconstruction is generated by adding the residue to the prediction of current block.

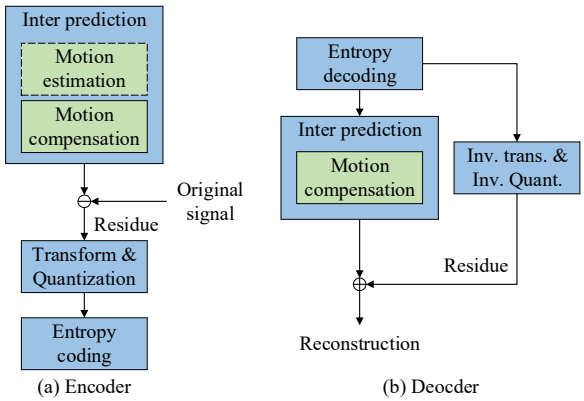

Fig. 2. Inter prediction in HEVC.

Table 1. The configuration of VRCNN [9]

Layer	Layer 1	Layer 2		Layer 3		Layer 4
Conv.module	conv1	conv2	conv3	conv4	conv5	conv6
Filter size	5x5	5x5	3x3	3x3	1x1	3x3
#filters	64	16	32	16	32	1
#parameters	1600	25600	18432	6912	1536	432
Total parameters	54512					

2.2. Review of VRCNN

In [9], VRCNN is designed to perform artifact reduction for HEVC. In the following, we will briefly review the overall architecture of VRCNN.

In VRCNN, there are four fully convolutional layers and variable filter size is adopted in the second layer and third layer. The outputs of different-sized filters are concatenated to be fed into the next layer. The first and the last layers of VRCNN do not use variable filter size. The recently developed residue learning technique [16] is integrated into VRCNN. That is, the output of the last layer is added back to the input.

3. NEURAL NETWORK BASED INTER PREDICTION

In this section, we will introduce the proposed neural network based inter prediction (NNIP) algorithm for HEVC. First, we concentrate on the network architecture of NNIP. Then we introduce the training strategy for the network. Finally, we integrate NNIP in HEVC to improve the compression performance of HEVC.

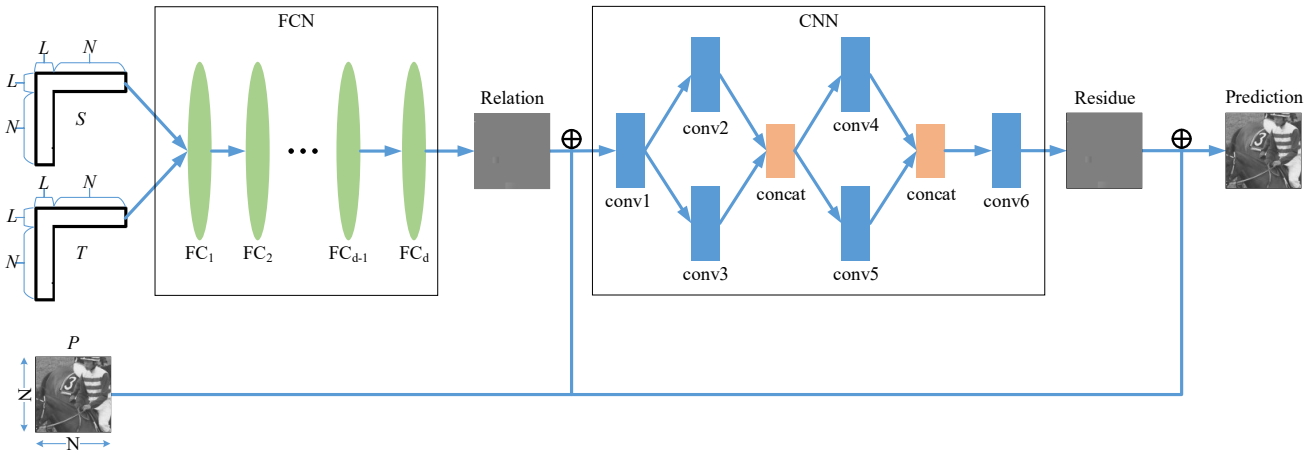

Fig. 3. The network architecture of NNIP.

3.1. Network architecture of NNIP

The network architecture of NNIP is shown in Fig. 3. It consists of two networks, namely a fully-connected network (FCN) and a convolutional neural network (CNN). There are three inputs for the network, including the spatial neighboring pixels, the temporal neighboring pixels and the prediciton of current block. They are represented by S, T, and P respectively in Fig. 4. S and T are used as the inputs for FCN. The output of FCN is added to P. The result is used as the input of CNN. The output of the network is the improved prediction of current block.

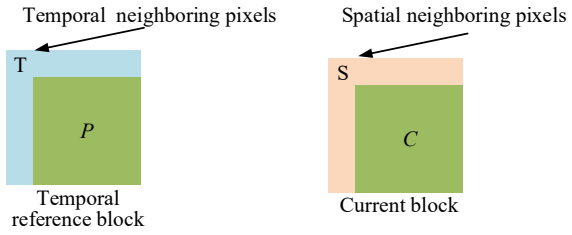

Fig. 4. Inputs of the network for NNIP.

In FCN, for a predicted block with size of $N \times N$, we use S and T as the inputs, shown in Fig. 3. As shown in Fig. 3, FC_i denotes the ith fully connected layer, and the depth of FCN is indicated by d. Denote the input by Y, the output of layer i by $G_i(Y)$, and the final output of FCN by $G(Y) = G_d(Y)$, then the network can be represented as follows:

$$\begin{aligned} G_1(Y) &= f(W_{1,F} \cdot Y + B_{1,F}), i = 1 \\ G_i(Y) &= f(W_{i,F} \cdot G_{i-1}(Y) + B_{i,F}), 1 < i \leq d \end{aligned} \tag{1}$$

where $W_{i,F}$ and $B_{i,F}$ are the weights and biases parameters of layer i in FCN, $\cdot$ denotes inner product. $f()$ is a non-linear mapping function, and the parameter rectified linear unit (PReLU) [17] is used as the non-linear mapping function in this paper.

For the first layer ($i = 1$), FC_1 is $4NL + 2L^2$-dimensional. For the hidden layers ($1 < i < d$), is K-dimensional, the output of the $(i-1)th$ layer. For the last layer ($i = d$), is N^2-dimensional, which is reshaped into the output, which is the relation block with size of $N \times N$.

As mentioned above, the output of FCN is added to P, which is the prediction of current block shown in Fig. 3. Then the result is used as the input of CNN. In this paper, VRCNN is adopted in this component. Detailed feature map numbers of each layer are shown in Table 1. Similar to FCN, the network of CNN can be represented as follows:

$$\begin{aligned} F_1(Y) &= g(W_{1,C} * Y + B_{1,C}), i = 1 \\ F_i(Y) &= g(W_{i,C} * F_{i-1}(Y) + B_{i,C}), 1 < i < d' \end{aligned} \tag{2}$$

where $W_{i,C}$ and $B_{i,C}$ are the weights and biases parameters of layer i in CNN, $*$ denotes convolution. $g()$ is a non-linear mapping function. d' equals to 4 in VRCNN.

The recently developed residue learning technique [16] is used in VRCNN. That is, the output of the last convolutional layer is added back to the input, which can be represented as follows:

$$F(Y) = W_{4,C} * F_3(Y) + B_{4,C} + Y \tag{3}$$

where $W_{4,C}$ and $B_{4,C}$ are the weights and biases parameters of last convolutional layer in VRCNN. $F_3(Y)$ denotes the output of 3th convolutional layer.

Similarly, the residue learning technique is also used in the proposed network to accelerate the speed of training. However, different from that in VRCNN, the final output is the sum of the output of CNN and P, denoting current block. This is represented as follows:

$$F(Y) = W_{4,C} * F_3(Y) + B_{4,C} + P \tag{4}$$

The parameters in each layer of CNN are the same as those in VRCNN. Zero padding is used for each convolution layer to ensure the same size of input and output blocks. Thus, the output of the overall network is a block with size of $N \times N$, which is the improved prediction of current block.

3.2. Training strategy

In this section, we will introduce the training strategy for the proposed network of NNIP, mainly focusing on training data generation and hyper parameters setting.

Denote (x_i, y_i) as each training sample, where x_i represents the inputs of the network including S, T, and P, and y_i denotes the label of P, which is the original signal of current block. To generate the training data, we first compress three video sequences in HEVC common test condition (*BasketballDrive*, *BQMall*, and *BlowingBubbles*) by using HM 16.9 with low delay P (LDP) configuration [18]. All frames of these three sequences are encoded with different quantization parameters (QP = 22, 27, 32, and 37). Second, we extract x_i from the compressed bitstreams and extract y_i from the original video sequences.

Learning the entire mapping F from the inputs to the enhanced prediction of current block needs to estimate the parameters. Specifically, given a collection of n training instances, where the ith instance consists of the inputs x_i and the original block y_i, we use the mean squared error (MSE) to minimize the following loss function:

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} ||F(x_i|\Theta) - y_i||^2 \tag{5}$$

where $\Theta = \{W_{j,F}, B_{j,F}, W_{k,C}, B_{k,C}\}, 1 \le j \le d, 1 \le k \le d'$, n is the total number of training samples.

We train our proposed network of NNIP by using the deep learning framework caffe on a NVIDIA GeForce GTX 1080 GPU. The loss function is minimized by using the first-order gradient based optimization Adam [19]. We adopt a batch-mode learning method with a batch size of 64. The momentum of Adam optimization is set to be 0.9 and the momentum2 is set to be 0.99. The depth of FCN d is set to be 4 in this paper. The dimension of hidden layers, K is set to be twice dimension of the input layer in FCN. The external lines L is set to be 4 in this paper. The base learning rate is set to decay exponentially from 0.1 to 0.0001, changing every 40 epochs. Thus, the training takes 160 epochs in total. We train the model for QP = 37 by using the above base learning rate. The models for other QPs (22, 27, and 32) are fine-tuned from the model of QP = 37. When fine tuning, the base learning rate is 0.001. In addition, we train different models for different sizes of CU, which varies from 8×8 to 64×64. Therefore, there are 16 models in total for the proposed network of NNIP.

3.3. Integration in HEVC

To evaluate the proposed NNIP algorithm, we integrate NNIP in HEVC to improve the compression performance of HEVC. NNIP is used to improve the accuracy of inter prediction, which is located after conventional inter prediction of HEVC. The simplified diagram is shown in Fig. 5.

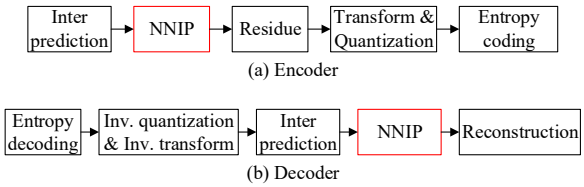

Fig. 5. The simplified diagram of NNIP integrated in HEVC.

As shown in Fig. 5 (a), during the process of encoding, NNIP is used after inter prediction. Specially, the spatial neighboring pixels, the temporal neighboring pixels, and current predicted block are inputted into the network of NNIP. Then the output of the network is the improved predicted block. As shown in Fig. 5 (b), during the process of HEVC decoding, NNIP is also used after inter prediction to generate the improved prediction of current block.

The proposed NNIP algorithm can improve the accuracy of inter prediction. First, the relation between the spatial neighboring pixels and the temporal neighboring pixels is learned by FCN. Then the result is added to the prediction current block, which is further improved by CNN. In HEVC, only the luma component is processed by NNIP. The proposed NNIP algorithm is only applied to inter/merge/skip 2N×2N mode in this paper. However, NNIP may hamper the compression performance of inter prediction when blocks have been predicted well. Therefore, a CU level flag is set to indicate whether NNIP is used by using rate distortion optimization.

4. EXPERIMENTAL RESULTS

In this section, extensive experiments are conducted to evaluate the performance of the proposed NNIP algorithm. First, experimental settings are given in Section 4.1. Then the coding performance are evaluated in Section 4.2. Finally, computational complexity is discussed in Section 4.3.

4.1. Experimental settings

NNIP is integrated in HM 16.9, which is the reference software of HEVC. The experiments follow the common test conditions defined in [18]. LDP setting configuration is simulated to demonstrate its performance. Total 18 sequences with 8 bit depth and 64 frames are encoded in our experiments, including Class A (4Kx2K), B (1080P), C (WVGA), D (QWVGA),

Table 2. The BD-rate of NNIP for luma component compared to HM 16.9

Class	Resolution	Sequence	BD-Y
Class A	2560x1600	*Traffic*	-1.5%
		PeopleOnstreet	-0.6%
Class B	1920x1080	*Kimono*	-1.9%
		ParkScene	-0.3%
		Cactus	-2.3%
		*BasketballDrive**	-3.8%
		BQTerrace	-8.6%
Class C	832x480	*BasketballDrill*	-1.3%
		*BQMall**	-2.2%
		PartyScene	-0.7%
		RaceHorses	-0.6%
Class D	416x240	*BasketballPass*	-0.9%
		BQSquare	-1.3%
		*BlowingBubbles**	-0.7%
		RaceHorses	-0.6%
Class E	128x720	*FourPeople*	-1.5%
		Johny	-2.0%
		KristenAndSara	-2.1%
Average			-1.7%

and E (720P). QP used in our experiments varies among 22, 27, 32, and 37. Computer with Intel i7-6700 3.4GHz Quad-core processors with 64GB memory and Microsoft Windows Server 2012 R2 operating system is used. Both HM 16.9 and the proposed algorithm are compiled with Microsoft Visual Studio 2013. When integrated in HEVC, the network of NNIP is processed with GPU version Caffe [20].

4.2. Coding performance

We compute the BD-rate [21] to evaluate the coding performance of NNIP. Table 2 shows the coding performance of the proposed algorithm compared to HM 16.9, in which the negative number indicates bitrate saving and the positive number indicates bitrate increasing. As shown in Table 2, the average coding gain is about 1.7% (up to 8.6%) for luma component, which demonstrates the efficiency of the proposed NNIP method.

As shown in Table 2, coding gains can be achieved for all test sequences. The coding gain changes largely for different sequences, which means that the proposed NNIP algorithm is affected by the contents of the video sequences. The proposed NNIP algorithm can achieve better compression performance for video sequences with high motion or rich texture, such as *BasketballDrive*, *BQTerrace*, and *BQMall*. In addition, although we use three HEVC sequences for training, denoted by * in Table 2, the coding gains for these sequences are not higher than others obviously. In the future, we will extend the training set with excluding the testing sequences to evaluate the proposed NNIP algorithm.

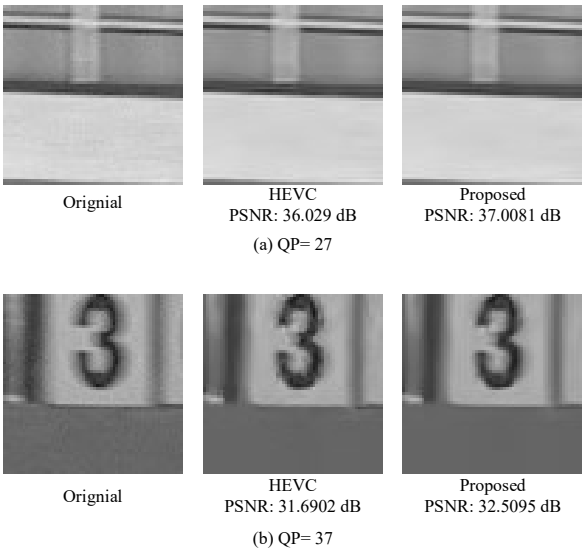

Fig. 6. Visual quality of the proposed NNIP algorithm.

To further evaluate the coding performance of the proposed NNIP algorithm. We also compare the visual quality of NNIP with the conventional inter prediction in HEVC. As shown Fig. 6, the proposed NNIP algorithm can improve the accuracy of inter prediction both in QP = 27 and QP = 37.

4.3. Computational complexity

The encoding and decoding complexities of the proposed NNIP algorithm are shown in Table 3. The computational complexity is evaluated by time increasing, which is denoted as follows:

$$\Delta T = \frac{T_p - T_o}{T_o} \times 100\% \tag{6}$$

where ΔT denotes the time increasing for encoding or decoding. T_o and T_p denote the encoding (decoding) time of HM 16.9 and the proposed NNIP method respectively. ΔT_{enc} and ΔT_{dec} denote the time increasing of encoding and decoding in Table 3 respectively.

As shown in Table 3, the encoding time increasing is about 3444% on average and the decoding time increasing is about 2022% on average. The high complexity mainly comes from two reasons. The first reason is that rate distortion optimization must be done for all sizes of CU and each mode of

Table 3. The computational complexity of NNIP

	ΔT_{enc}	ΔT_{dec}
Class A	3273%	1700%
Class B	3314%	3301%
Class C	2479%	2416%
Class D	2842%	1578%
Class E	5310%	1113%
Average	3444%	2022%

inter prediction. The second reason is that the forward operation of the proposed network. The computational complexity will be investigated in our future work.

5. CONCLUSION

In this paper, we propose a neural network based inter prediction method for HEVC. A novel architecture by combing fully connected network and convolutional neural network is proposed, in which the spatial and temporal neighboring pixels and the prediciton of current block are as the inputs. It can improve the accuracy of inter prediction in HEVC. Experiments show that the proposed NNIP algorithm can achieve average 1.7% (up to 8.6%) BD-rate reduction. Application for other inter modes will be investigated in the future.

6. REFERENCES

[1] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[3] P. Yin, A. M. Tourapis, and J. Boyce, "Localized weighted prediction for video coding," in *2005 IEEE International Symposium on Circuits and Systems*, May 2005, pp. 4365–4368 Vol. 5.

[4] S. Jeong and H. Park, "Offset compensation method for skip mode in hybrid video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1825–1831, Oct 2014.

[5] N. Zhang, Y. Lu, X. Fan, R. Xiong, D. Zhao, and W. Gao, "Enhanced inter prediction with localized weighted prediction in hevc," in *2015 Visual Communications and Image Processing (VCIP)*, Dec 2015, pp. 1–4.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, 2014, pp. 184–199.

[7] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 576–584.

[8] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of jpeg-compressed images," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2764–2772.

[9] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in hevc intra coding," in *MultiMedia Modeling. MMM 2017. Lecture Notes in Computer Science*. Springer, 2017, pp. 28–39.

[10] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for hevc," in *2017 Data Compression Conference (DCC)*, April 2017, pp. 410–419.

[11] R. Yang, M. Xu, and Z. Wang, "Decoder-side hevc quality enhancement with scalable convolutional neural network," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 817–822.

[12] W. S. Park and M. Kim, "Cnn-based in-loop filtering for coding efficiency improvement," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, July 2016, pp. 1–5.

[13] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," in *2017 Visual Communications and Image Processing (VCIP)*. arXiv preprint arXiv:1709.08462, 2017.

[14] J. Li, B. Li, J. Xu, and R. Xiong, "Intra prediction using fully connected network for video coding," in *2017 Visual Communications and Image Processing (VCIP)*, 2017.

[15] Y. Li, H. Li, L. Li, F. Wu, H. Zhang, and H. Yang, "Convolutional neural network-based block up-sampling for intra frame coding," arXiv preprint arXiv:1702.06728.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1026–1034.

[18] F. Bossen, "Common test conditions and software reference configurations," JCTVC-L1100, Jan 2013.

[19] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*. 2014, pp. 675–678, ACM.

[21] G. Bjϕntegaard, *Improvements of the BD-PSNR Model*, document VCEG-AI11, ITU-TVideo Coding Experts Group (VCEG), Heinrich-Hertz-Institute, Berlin, Germany, July 2008.