An Integrated CNN-based Post Processing Filter For Intra Frame in Versatile Video Coding

Mingze Wang*, Shuai Wan†, Hao Gong*
Northwestern Polytechnical University, Xi'an, China
E-mail: *{roronoa, gong_h}@mail.nwpu.edu.cn, †swan@nwpu.edu.cn
Yuanfang Yu, Yang Liu
Guangdong OPPO Mobile Telecommunications Corp., Ltd, China

***Abstract*— Versatile Video Coding (H.266/VVC) standard achieves up to 30% bit-rate reduction while keeping the same quality compared with H.265/HEVC. To eliminate various coding artifacts like blocking, blurring, ringing, and contouring effects, etc., three in-loop filters have been incorporated in H.266/VVC. Recently, convolutional neural network (CNN) has attracted tremendous attention and achieved great success in many image processing tasks. In this paper, we focus on CNN-based filtering in video coding, where a single model solution for post-loop filtering is designed to replace the current in-loop filters. An architecture is proposed to reduce the artifacts of video intra frames, which take advantage of useful information such as partitioning modes and quantization parameters (QP). Different from existing CNN-based approaches, which generally need to train different models for different QP and only suitable for luma component, the proposed filter can well adapt to different QP, i.e. various levels of degradation of frames, and all components (i.e., luma and chroma) are jointly processed. Experiment results show that the proposed CNN post-loop filter not only can replace the de-blocking filter (DBF), sample adaptive offset (SAO) and adaptive loop filter (ALF) in H.266/VVC, and also outperforms them, leading to 6.46%, 10.40%, 12.79% BD-rate savings for Y, Cb and Cr, respectively, under all intra configuration.**

I. INTRODUCTION

As the image and video coding standards evolves, e.g., JPEG[1], H.264/AVC[2], H.265/HEVC[3], H.266/VVC[4], the artifacts, distortion, noise, blurring, and loss of critical details in high frequency are still inevitable while the compression efficiency is getting higher. The most noticeable and distinct distortion that affects the subjective quality is the discontinuity of pixel value at the boundary of the block, forming an obvious line, which is called blocking artifacts. This is due to the block-based video compression framework has been widely used and almost all the operations in the process of compression such as prediction, transformation, quantization are conducted on the block level. In addition to the blocking artifacts, due to the loss of high-frequency information, there will be ringing effect and fuzzy effect on the picture. The lower the bitrate is, the more obvious these distortion will be. Although the distortion is irreversible and cannot be completely eliminated, it can be reduced. Therefore, the decoded frames are generally filtered in-loop to alleviate these degradation and improve the quality before used as reference frame of the current frame or the output frame.

The in-loop filters in current video coding standard are designed for different artifacts, for example, de-blocking filter (DBF)[5] for blocking artifacts, sample adaptive offset (SAO)[6] for ringing artifacts, adaptive loop filter (ALF)[7] for distortion. However, the current processing technique may not sufficiently remove the compression artifacts in low bitrate encoding, and additional binary stream need to be transmitted due to the flags and filter coefficients. On the other hand, convolutional neural network (CNN)-based filtering methods [10]~[19] have achieved good performances. A problem of the state-of-the-art CNN-based filtering methods lies in lack of generalizations on various qualities caused by different quantization parameters (QP) during coding process, and limitation on filtering both the luma and the chroma components. To this end, we propose a post processing filter as a total solution to replace filters in the current standard. The contributions of this paper are summarized as follows.

- We utilize the QP as a prior knowledge. Adding QP information as an input makes the network adaptive to multi-quality reconstructions with a single set of parameters.
- In order to reduce the blocking artifacts and replace the de-blocking filter, we use the partition information which indicates the location of distortion at block boundaries to guide the quality enhancement.
- We design a three-branch network structure to generate different range of compensation for three components.
- The proposed network is designed for but not limited to H.266/VVC. It can be well adapted to other video coding framework, e.g., AVS3, H.265/HEVC and etc.

H.266/VVC has more intra prediction modes than H.265/HEVC, i.e., the number of directional intra modes has been extended from 33 to 65 [8], and has a new prediction mode introduced as cross-component linear model (CCLM) [9] where the the chroma samples are predicted based on the reconstructed luma samples of the same Coding Unit (CU) by using a linear model. Therefore, efficient filtering for intra frames in H.266/VVC is in great need. The rest of this paper is organized as follows. Section II provides a brief review of in-loop filtering in video coding and related work of CNN-based filtering. The proposed network and method are

described in section III. Experimental results are reported in section IV. In section V, we conclude this paper.

II. Related Work

In this section, we briefly review the related work about in-loop or post-loop filters in image/video compression. Firstly, the filters in current video coding standard will be introduced. And then we discuss the CNN-based method in detail.

A. In-loop Filters in Video Coding

Widely used video coding standards like H.264/AVC, and H.265/HEVC, as well as H.266/VVC under formulation, are all jointly developed by Video Coding Experts Group (VCEG) of ITU-T and Moving Picture Experts Group (MPEG) of ISO/IEC. While representing the state-of-the-art at their time, the lossy compression framework leads to various kinds of artifacts. To reduce those kinds of degradations in the coding process, the following three in-loop filtering algorithms have been adopted in video coding to enhance the quality of decoded frames for better reference.

Due to block-based coding, the discontinuity often appears along the block boundaries. This becomes more unpleasant when a noticeable line formed under low bit rates. The blocking artifacts mainly comes from two aspects. Firstly, after quantization and inverse quantization, the difference of pixel values between blocks is magnified. Secondly, in the motion estimation process, the selected reference blocks are sometimes copied from different positions of different reference frames, and these matching blocks are not absolutely accurate, so the pixel value discontinuity will be generated on the boundary. The de-blocking technology [5] first obtains the boundary strength according to the coding parameters of the blocks on both sides of the boundary and the change of pixel value, and then carries on the corresponding filtering to the boundary that needs to be filtered using low-pass filters. Although de-blocking filters can reduce the blocking artifacts efficiently by smoothing the pixels on boundary, but the inner pixels within the block remains ignored. The other kind of distortions need to be further reduced by other methods.

Sample Adaptive Offset (SAO) [6] is a filter targeted for the ringing distortion which is caused by the loss of high frequency detail during the transform and quantization. The reconstructed pixel value fluctuates up and down around the real pixel value, forming a wavy distortion. SAO algorithm divide the reconstructed pixel value into categories according to its characteristics, and then add negative values to crest pixel and positive values to valley pixel for compensation.

Adaptive loop filter (ALF) [7] is based on Wiener filter aims to minimize the distortion between the reconstructed frame and the original one. The filter coefficients are trained at the encoder side and transmit to the decoder.

B. CNN-based quality Enhancement and filtering

Recently, inspired by the great success in high-level vision tasks, CNN was also adopted for low-level vision tasks to find a nonlinear mapping function from the degraded image to the desired one, such as super-resolution, image restoration, image quality enhancement, image de-noising, etc. Since CNN has powerful nonlinear fitting ability, it can achieve the state-of-the-art performance for the ill-posed image restoration problems.

The first work that used the CNN for artifacts reduction is the ARCNN [10] which inspired by super-resolution network SRCNN [11]. It stacked four convolution layers with the mean squared error (MSE) loss function and boosted the restoration quality of JPEG decoded images. Authors in [12] first applied the idea of residual learning into reconstruction, and designed the mean square error loss function based on Sobel operator, which recovered the high-frequency details well. The IFCNN in [13] trained two models with low QP(20~29) and high QP(30~39) during training, which are applicable to different levels of degradation. The network structure designed in [14] includes both the processing module of pixel domain and the processing module of DCT domain, making full use of the characteristics of both domains. The VRCNN proposed in [15] replaced DBF and SAO in H.265/HEVC, which could save 4.6% BD-rate. A very complex network topology is used in [16], which adopted the weighted multi-scale loss generated by multiple outputs. In [17], inputs include current frame and adjacent frames, which enables the network to capture not only spatial relations, but also temporal relations between frames for joint de-noising. In [18], in order to solve the problem of the small reception field of convolution kernel, pixel rearrangement algorithm is adopted to sample the image and multiply the number of channels by 4 times, so that the convolution kernel of the same size can process a larger area without losing any information. In [19], considering that the green component of the image is usually the one with higher quality of reconstruction, the green component is firstly enhanced, and then the results are fused with the red/blue component to guide its enhancement process.

III. Proposed Single CNN Model

In this section, we first discuss the way to make a single model being capable of handling various level of degradation. Then we describe the proposed multiple-input CNN.

A. Adaptive Generalization For Different QP

Different quantization parameters (QP) will lead to diverse reconstructed video frame quality. The larger the quantization parameter is, the greater the distortion will be, and the larger the distribution range of the compensation value between the reconstructed pixel and original pixel will be. The value of QP presents an approximate linear relationship with the distribution range of compensation values. For a network with a global residual connection, the output of the network should be as close to the compensation value as possible before connecting the input value. If a set of network parameters are required to adapt to the compensation values of different distribution ranges, the network should obtain this prior

information in order to better filter the input with different quality.

For the prior information QP, we construct a feature map named as QPmap which is as the same size of input size, filled with the normalized QP value of the current frame as in (1), and concatenated with other feature maps. The *MAXQP* is set to 63 in H.266/VVC. Since each feature map of the convolution layer will weights all the feature maps of the previous layer, this method guides the network to convert different QP values into compensation values of different amplitudes at corresponding positions through convolution in subsequent operations.

$$QP\text{map}(x,y) = \frac{QP}{MAXQP},\ \text{x} = 1,...,\text{W}\ \ \text{y} = 1,...,\text{H} \qquad (1)$$

B. Partition Information Fusion

Since the blocking artifacts mainly appear along the boundary of coding unit (CU) blocks, we can use the informative partitioning structure to effectively guide the quality enhancement process.

In H.265/HEVC and H.266/VVC video coding standard, a frame will be firstly divided into a sequence of coding tree units (CTUs). But different from the single quaternary-tree partitioning method in H.265/HEVC, a quad-tree with nested multi-type tree using binary and ternary splits segmentation structure [8] is adopted as the initial new coding feature of H.266/VVC. A CTU is first partitioned by a quaternary tree structure. Then the quaternary tree leaf nodes can be further partitioned by a multi-type tree structure which has four splitting types as shown in Fig 1.

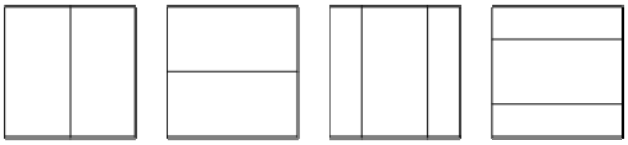


Fig. 1. Multi-type tree structure. (a) vertical binary splitting. (b) horizontal binary splitting. (c) vertical ternary splitting. (d) horizontal ternary splitting.

In order to feed CU partitioning information into network, we could construct a feature map named CUmap, with the positions of the boundary are filled by 1 and other positions by 0.5 as shown in Fig. 2. It indicates areas and boundaries that require significant compensation and correction. Additionally, it should be noted that there are two partition trees for I frame, one for luma component, the other shared by two chroma components.

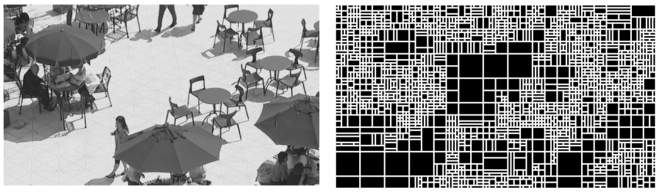


Fig. 2. The Y component and its CUmap in sequence "BQSquare" of QP 37.

C. Processing Block

The basic processing block in the proposed network is composed of convolutional layers, locally skip connection and SE (Squeeze-and-Excitation) [20] operation which is known as a method of weighting each convolutional layer adaptively. It is able to exploit the complex relationship between different channels and generate a weighting factor for each channel. It is shown in Fig. 3.

Given a feature map X with shape H*W*C, the processing operation includes several steps as following.

a. Two convolutional layers with Rectified Linear Unit (ReLU) [21] between them are conducted firstly.

$$Y_1 = \text{Re}\,LU(W_1 * X + b_1) \qquad (2)$$

$$Y_2 = W_2 * Y_1 + b_2 \qquad (3)$$

b. Each channel is squeezed to a single numeric value using Global Average Pooling (GAP) according to (4)

$$Y_3(i) = \frac{1}{WH}\sum_{m=1}^{W}\sum_{n=1}^{H} Y_2(m,n,i) \quad \text{i=1,...,C} \qquad (4)$$

c. A fully connected layer followed by a ReLU function adds the necessary nonlinearity. Its output channel complexity is also reduced by a certain ratio r which is set to be 4 in this paper.

$$Y_4 = \text{Re}\,LU(W_4 Y_3 + b_4) \qquad (5)$$

d. A second fully connected layer followed by a sigmoid activation gives each channel a smooth gating ratio ranged in [0,1].

$$Y_5 = sigmoid(W_5 Y_4 + b_5) \qquad (6)$$

e. Each channel of Y_2 is scaled by the gating ratio by (7)

$$Y_6(m,n,i) = Y_2(m,n,i) \times Y_5(i)$$
$$\text{m=1,...,W}\ \ \text{n=1,...,H}\ \ \text{i=1,...,C} \qquad (7)$$

f. At last, if the number of input channels is the same as that of the output channels, a skip connection will be added from the input into the output directly to learn the residual, it can also benefit fast convergence. Otherwise, there is no skip connection.

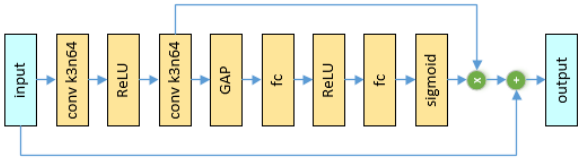


Fig. 3. The basic processing block

D. Network Structure

Unlike the RGB color format, the components in YUV format are diverse from each other in many aspects. For example, the pixel values of luma component have much wider dynamic range than that of chroma components, and the average reconstruction quality of the three components is also

significantly different, leading to difficulty in diverse filtering. To address these problems, we design a two-stage three-branch CNN network to process all three components simultaneously as shown in Fig. 4.

At the first stage, we upsample the U/V components to align the matrix's sizes since the width and height of U/V component have only half size of that of Y component in YUV4:2:0 format. Then the 3-channel input will be processed by several basic processing block. The QPmap is concatenated at this stage.

And in the second stage, the main pipeline will be split into three branches, each branch is for one component and fused by its own CUmap. Then several basic processing block is conducted for each branch to generate the final residual image, followed by global skip connection for reconstructed image.

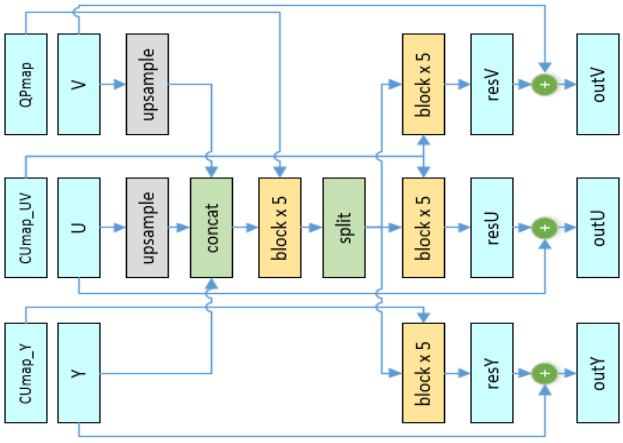


Fig. 4. Network structure.

data through H.266/VVC reference software VTM-3.0 [8] under all intra configuration using QP values in {22, 27, 32, 37}. We convert the picture from PNG format to YUV4:2:0 format, and then collect the distortion image before in-loop filters as the input and the original image as ground truth. For each image, we divide it into 48×48 patches without overlapping.

Training settings. We implement the proposed model using TensorFlow [23]. During training, we use batch size of 256, and start with a learning rate of 1e-3, decay the learning rate with 2 every 100 epochs.

Loss function. We adopt MSE criterion between the output of the network and the ground truth image. In addition, since the improvement of the Y component is more important than that of the U/V component, we assign more weight to the Y component. The final loss function is as (8).

$$loss = 8\times MSE(Y,\hat{Y}) + MSE(U,\hat{U}) + MSE(V,\hat{V})$$
$$= 8\times\frac{\left\|Y-\hat{Y}\right\|_2^2}{WH} + \frac{\left\|U-\hat{U}\right\|_2^2}{W/2\times H/2} + \frac{\left\|V-\hat{V}\right\|_2^2}{W/2\times H/2} \quad (8)$$

Test conditions. We integrate the final model into VTM-3.0 using TensorFlow frozen proto buffer files and TensorFlow C++ API. The experiments are conducted under H.266/VVC common test condition [24] with all intra configurations but the DBF, SAO, and ALF are disabled.

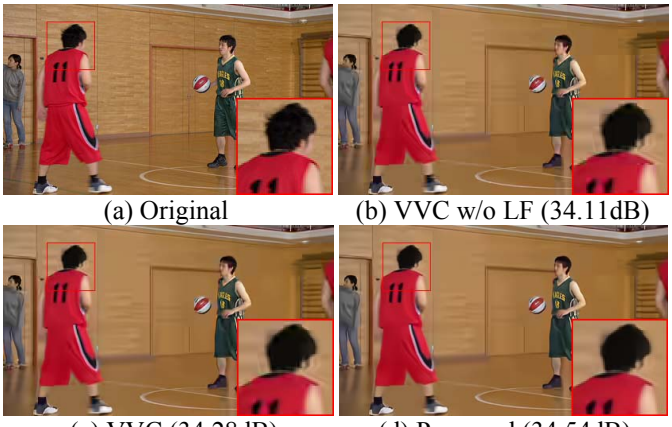
Comparison. Table I compares the overall BD-rate [25] saving of different methods over the H.266/VVC reference software VTM-3.0 [8]. Five methods are compared: (1) H.266/VVC anchor with DBF, SAO and ALF enabled; (2) Method in ref. [26] located between DBF and SAO; (3) Method in ref. [27] which only replace DBF and SAO, but ALF is enabled; (4) Method in ref. [28] located between DBF and SAO; (5) Our method with DBF, SAO and ALF all disabled. From Table I, we can observe that the proposed network achieves the best performance overall the compared methods. It can obtain 6.46%, 10.40%, 12.79% BD-rate reduction on luma and two chroma components, respectively. For subjective quality comparison, some example images are shown in Fig. 5. The results demonstrate that the proposed network effectively removes the blocking artifacts and especially ringing artifacts, and enhances the visual quality as well.

TABLE I
THE RATE-DISTORTION PERFORMANCE COMPARISON WITH OTHER METHODS IN AI CONFIGURATION. (ANCHOR: VTM-3.0)

method	network in [26]			network in [27]			network in [28]			ours		
filter chain	DBF+NN+SAO+ALF			NN+ALF			DBF+NN+SAO+ALF			**NN**		
component	Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class B	-0.63%	-0.02%	-0.02%	-1.22%	-1.18%	-1.23%	-1.33%	-1.57%	-2.12%	**-4.48%**	**-11.00%**	**-11.15%**
Class C	-1.72%	-1.33%	-1.89%	-2.25%	-2.75%	-4.21%	-2.85%	-1.61%	-1.45%	**-6.40%**	**-9.07%**	**-12.64%**
Class D	-2.18%	-1.39%	-1.64%	-2.70%	-2.55%	-4.12%	-3.57%	-1.74%	-1.94%	**-6.67%**	**-12.12%**	**-16.35%**
Class E	-1.68%	0.01%	0.01%	-2.69%	-2.11%	-2.33%	-3.65%	-0.84%	-0.86%	**-8.30%**	**-9.41%**	**-11.03%**
Overall	-1.55%	-0.68%	-0.89%	-2.22%	-2.15%	-2.97%	-2.85%	-1.44%	-1.59%	**-6.46%**	**-10.40%**	**-12.79%**

IV. EXPERIMENTS

Dataset. Since the proposed network is for intra coding, we could use large lossless picture datasets instead of video datasets. DIV2K dataset [22] which is consist of 900 2K resolution PNG pictures (800 images for training, and 100 images for validation) is used to derive training and validation



(a) Original (b) VVC w/o LF (34.11dB)

(c) VVC (34.28dB) (d) Proposed (34.54dB)

Fig. 5. Visualization quality comparison results on the sequence "BasketballPass" of QP 37.

V. CONCLUSIONS

In this paper, a CNN-based post-processing method is proposed which serves as an integrated solution for filtering to replace the in-loop filters. It achieves good performance by taking advantages of information about quantization parameters and partitioning structure, where all three components are used as inputs, leading to better filtering quality for wide range of degradation levels.

REFERENCES

[1] G. K. Wallace, "The JPEG still picture compression standard," in IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992.

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

[3] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[4] B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 3)," document JVET-L1001, 12th JVET meeting: Macao, CN, 3–12 Oct. 2018.

[5] A. Norkin et al., "HEVC Deblocking Filter," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1746-1754, Dec. 2012.

[6] C. Fu et al., "Sample Adaptive Offset in the HEVC Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1755-1764, Dec. 2012.

[7] C. Tsai et al., "Adaptive Loop Filtering for Video Coding," in IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 934-945, Dec. 2013.

[8] J. Chen, Y Ye, and S. H. Kim, "Algorithm description for Versatile Video Coding and Test Model 3 (VTM 3)", document JVET-L1002, 12th JVET meeting: Macao, CN, 3–12 Oct. 2018.

[9] K. Zhang, J. Chen, L. Zhang, X. Li and M. Karczewicz, "Enhanced Cross-Component Linear Model for Chroma Intra-Prediction in Video Coding," in IEEE Transactions on Image Processing, vol. 27, no. 8, pp. 3983-3997, Aug. 2018.

[10] C. Dong, Y. Deng, C. C. Loy and X. Tang, "Compression Artifacts Reduction by a Deep Convolutional Network," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 576-584.

[11] Dong, Chao, et al. "Learning a Deep Convolutional Network for Image Super-Resolution." european conference on computer vision (2014): 184-199.

[12] Svoboda, Pavel et al. "Compression Artifacts Removal Using Convolutional Neural Networks." CoRR abs/1605.00366 (2016): n. pag.

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

[14] Guo, Jun, and Hongyang Chao. "Building Dual-Domain Representations for Compression Artifacts Reduction." european conference on computer vision (2016): 628-644.

[15] Dai, Yuanying, Dong Liu, and Feng Wu. "A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding." conference on multimedia modeling (2017): 28-39.

[16] L. Cavigelli, P. Hager and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 752-759.

[17] C. Jia, S. Wang, X. Zhang, S. Wang and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, 2017, pp. 1-4.

[18] Tang, Zhimin and Linkai Luo. "Compression artifact removal using multi-scale reshuffling convolutional network." CVPR Workshops (2018).

[19] Cui, Kai and Eckehard G. Steinbach. "Decoder Side Image Quality Enhancement exploiting Inter-channel Correlation in a 3-stage CNN: Submission to CLIC 2018." CVPR Workshops (2018).

[20] J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7132-7141.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th international conference on machine learning (ICML), 2010, pp. 807–814.

[22] DIV2K, https://data.vision.ee.ethz.ch/cvl/DIV2K/

[23] Mart´ın Abadi, Ashish Agarwal, and Paul Barham et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[24] F. Bossen, J. Boyce, X. Li, and V. Seregin, K. Sühring, "JVET common test conditions and software reference configurations for SDR video," document JVET-L1010, 12th JVET meeting: Macao, CN, 3–12 Oct. 2018.

[25] Gisle Bjontegaard, "Calculation of average psnr differences between rd-curves," in ITU-T Q. 6/SG16 VCEG, 15th Meeting, Austin, Texas, USA, April, 2001, 2001.

[26] Y. Dai, D. Liu, Y. Li, and F. Wu, "AHG9: CNN-based in-loop filter proposed by USTC" document JVET-M0510, 13th JVET meeting: Marrakech, MA, 9–18 Jan. 2019.

[27] K. Kawamura, and S. Naito, "AHG9: A Result of Convolutional Neural Network Filter" document JVET-M0872, 13th JVET meeting: Marrakech, MA, 9–18 Jan. 2019.

[28] Y. Wang, Z. Chen, Y. Li, L. Zhao, S. Liu, and X. Li, "AHG9: Test Results of Dense Residual Convolutional Neural Network based In-Loop Filter" document JVET-M0508, 13th JVET meeting: Marrakech, MA, 9–18 Jan. 2019.