MULTI-MODAL/MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK BASED IN-LOOP FILTER DESIGN FOR NEXT GENERATION VIDEO CODEC

Jihong Kang[⋆], *Sungjei Kim*[†], *Kyoung Mu Lee*[⋆]

[⋆]Department of ECE, ASRI, Seoul National University, Seoul, Korea
[†] Korea Electronics Technology Institute, Seongnam-si, Korea

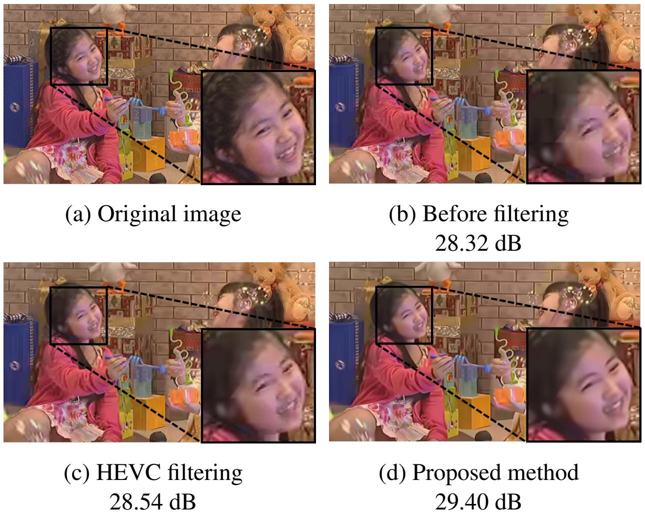
ABSTRACT

In this paper, we propose a novel in-loop filter design for video compression. Our approach aims to replace existing deblocking filter and SAO (Sample Adaptive Offset) of HEVC standard with multi-modal/multi-scale convolutional neural network (MMS-net). The proposed CNN architecture consists of two sub-networks of different scales. An input image is down-sampled first and restored through the lower scale network, then the output image from it is fed into higher scale network concatenated with the original input image. Moreover, to boost the restoration performance, the proposed architecture utilizes information resides in the coded sequence. Specifically, the compression parameters from coding tree units (CTU) are exploited as input to CNN, which helps to alleviate blocking artifacts on the reconstructed images. In the experiments, our method reduces the average BD-rate by 4.55% and 8.5%, respectively, compared with the conventional neural network based approach [1] and HEVC reference software HM16.7 [2] in 'All Intra - Main' configuration.

***Index Terms*—** In-loop filter, HEVC, CNN, video compression, image restoration

1. INTRODUCTION

The compression process of videos and images inherently causes distortion of the frame content. Compression technology has advanced toward preserving the quality of content while spending fewer bits for storing the compressed data. Among those technologies, the deblocking filter [3] and sample adaptive offset (SAO) [4] in HEVC standard [5] take an important role for removing visual artifacts such as blocking artifact, ringing artifact, blurring artifact and so on. As the name indicates, the deblocking filter is mainly responsible for removing blocking artifact which is caused by a block-based quantization process. The SAO, newly introduced in HEVC standard, compensates the other artifacts using additional offset values. These two filters restore both the subjective quality and the objective quality of the reconstructed image, but also contribute to enhancing the compression rate because the restored image is referenced to the prediction of other frames in the sequence.


(a) Original image

(b) Before filtering 28.32 dB

(c) HEVC filtering 28.54 dB

(d) Proposed method 29.40 dB

Fig. 1: Comparison of results between the proposed method and HEVC deblocking filter / SAO. PSNR is measured in the Y channel.

Recently, the success of convolutional neural network (CNN) in image classification has spread to many other research fields. In image restoration such as superresolution [6, 7, 8], deblurring [9, 10], and denoising [11], CNN already outperforms conventional non-learning based methods. From a machine learning perspective, we can think that CNN performs image restoration tasks by learning nonlinear mapping functions from distorted images to undistorted images using huge amounts of training data. Thus, although the distortion characteristics of the input images are different, the learning process of CNN is very similar among different image restoration tasks. In general, image pairs of distorted and undistorted images are fed into CNN as inputs and targets, and CNN learns how to restore the target image by removing the distortion of the input images. Likewise, the same process can be applied to learning the operation of the in-loop filter in video coding.

A few studies have begun to apply CNNs to replace the in-loop filters. Park and Kim [12] proposed the IFCNN for

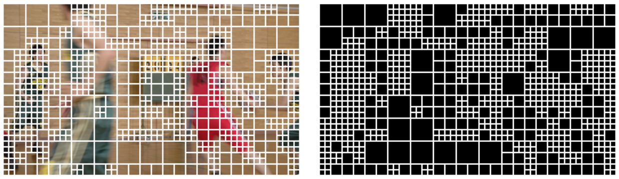

(a) CUs for an image (b) Corresponding CU image

Fig. 2: An example of formatting coding unit(CU) information. The boundary region (white) is set to 2 and the non-boundary region (black) is set to 1 in the implementation.

replacing SAO. IFCNN has 3 convolutional layers with a skip connection [7] between the input and the output. They showed the possibility of replacing existing in-loop filters with CNNs by reducing 1.6 - 2.8% of BD-rate. Dai et al. [1] proposed VRCNN as the replacement of both deblocking filter and SAO. VRCNN has total 4 stacks of convolutional layers with 2 different sizes of kernels in a single layer. VRCNN reported a promising result of average 4.6% bit-rate reduction. However, the CNN architectures of [1, 12] are too shallow and do not reflect recent improvements in CNN such as batch normalization [13] and residual networks [14, 15]. Moreover, these works only focused on applying CNNs to this problem naïvely without consideration of rich information inside compressed video.

A coded video has a wealth of information that directly or indirectly affects to the distortion of the frames such as the coding parameters (CPs), so we can leverage that information for restoration. For example, the hierarchical coding tree unit (CTU) information can exactly locate the boundaries of possible blocking artifacts. Exploiting this information as input to CNN can help the network precisely detect artifacts and recover them.

In this paper, we propose a novel in-loop filter design for HEVC based on multi-modal/multi-scale CNN (MMS-net) architecture. Our model has a fully convolutional neural network structure and works in an end-to-end manner. The structure of multi-scale CNN effectively enhances the restoration performance by coarse-to-fine restoration process. Also, the CTU information in the coded video guides the network to correctly detect and remove blocking artifacts. To better handle CTU information, we propose a method for converting CTU to a matrix form and a pre-network for matching different modal features.

2. MULTI-MODAL/MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK

2.1. Formatting Coding Parameters

Blocking artifacts are mainly caused by block-based compression mechanism. Thus, given the block partitioning information, the network can easily locate and eliminate blocking artifacts. In the HEVC standard, CTU is the basic processing unit for compression. A CTU is divided into quadtrees of coding units (CUs) recursively, and each CU is composed of predictive units (PUs) and transformation units (TUs). For simplicity, we utilize only CU and TU information of each frame as inputs for the CNN.

To provide the CU and TU information into CNN, the information should be converted into the proper form. CU and TU information should be able to provide each unit's position and size in the input image. In our implementation, we create a matrix of the input image size. Then, we assign the value of '2' to the positions corresponding to the outermost pixels of each CU and TU, and set the value of '1' for the non-boarder area in each matrix. Therefore, two matrices are generated for each frame. An example of CU encoding is visualized in Figure 2.

2.2. Multi-modality Adaptation Network

The most naïve approach for feeding the encoded CPs into the network would be concatenating the CP matrices along the channel axis of the input image. However, providing CP information to the network in this way may hamper the restoration process of the network, rather than improving the results due to different modalities. To effectively handle multi-modal information in a single network, we add a simple pre-network (called adaptation network) that transforms CP information space into image feature space. A combination of convolution layers and ReLU layers projects the CP matrices into a single channel feature map. Then, the CP feature map is element-wisely multiplied with the input image and concatenated as an additional channel to the input image. The structure of the adaptation network is shown in the upper left corner of Figure 3.

2.3. Model Structure

Recently, multi-scale image restoration architecture has been utilized in various image restoration researches. The multi-scale image restoration can be viewed as a hierarchical process in a multi-scale image space so that the restored image retains small details on finer scales as well as long-range dependencies on coarser scales. For example, Nah et al. [9] created three-level Gaussian pyramid images from a blurred image and used them as inputs to the CNNs corresponding to each scale for deblurring operations. The concept of coarse-to-fine restoration has proven useful in sharpening severely blurred images. Similarly, our proposed model has 2 sequential sub-networks of different scales ($K = 2$). The coarse-scale network recovers the degraded image from the half down-sized input image and the fine-scale network outputs the restored image by taking both the reconstructed image from the coarse-scale network and the original degraded image as inputs. Instead of resizing the input images from the outside of the network, in the coarse-scale network, we used

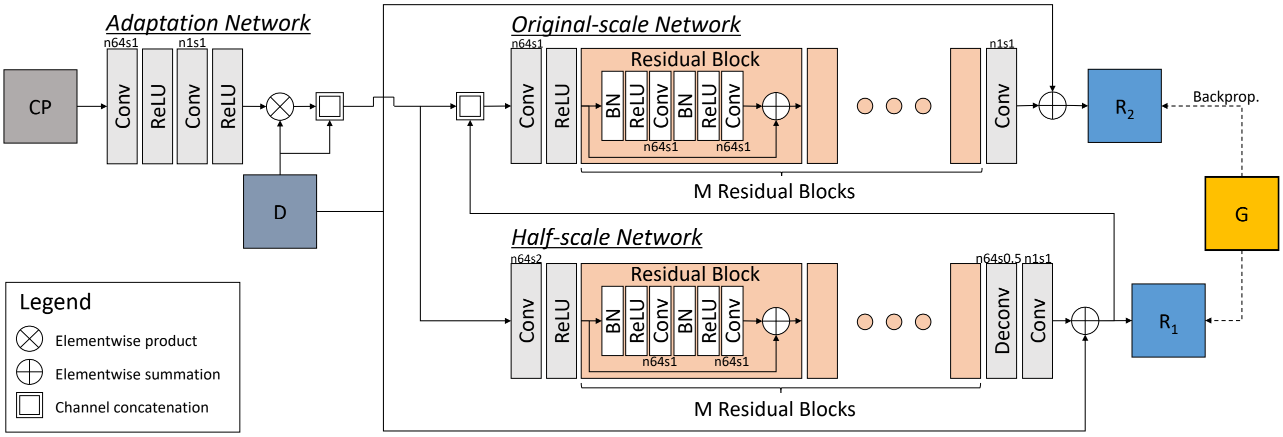


Fig. 3: Overall architecture of MMS-net. CP, D, R_k, G, and BN denote encoded coding parameters, a distorted image, a restored image from k-th scale path, a ground truth image, and a batch normalization layer. 'n $\underline{l}$ s $\underline{m}$' indicates a convolutional layer with $\underline{l}$ kernels and stride $\underline{m}$.

a convolutional layer with stride 2 for down-sampling and a deconvolutional layer for up-sampling. This interpolation structure embedded in the network simplifies the process of the entire system.

The sub-network structure of each scale is a modified version of the SRResnet [8]. The basic building unit of the network is residual blocks [15]. Each residual block contains two consecutive sub-modules consisting of a batch normalization layer, a ReLU layer, and a convolution layer. A skip connection between the input and output of each residual block directly propagates the input signal to the output, so the sub-modules with convolution layers learn residual features over the input signal. Similarly, a global skip connection [7] between inputs and outputs of each sub-network guides the network to generate the residual of restored image over the distorted image. We used M number of residual blocks for each scale path. All convolutional layers in the network use 3×3 kernels, except for the first convolutional layer of each scale's sub-network which uses 7×7 kernels. Padding is added to the boundaries of each convolutional layer to make the input and output dimensions same. The number of kernels and strides for each convolutional layer is described in Figure 3 with overall architecture.

Note that the proposed model has no fully-connected layer, thus, this is a fully convolutional neural network [16] which can generate any size of images as same as that of an input.

2.4. Multi-scale Loss

Loss on each sub-network is computed separately by the MSE criterion between the output of the sub-network and the ground truth image. Then, the overall loss is defined as follows:

$$L = \frac{1}{2wh} \sum_{k=1}^{K} \|R_k - G\|_2^2, \tag{1}$$

where R_k, G, K denote the output of the k-th sub-network, a ground truth image, and the number of sub-networks (scales), respectively. The loss is normalized by width w and height h of the input image. Note that the loss of the k-th scale sub-network is backpropagated through the network and accumulated with the loss of the $(k-1)$-th level in the proposed architecture.

2.5. Training

For training dataset, we used 28 HD sequences in YCbCr 420 color format from Xiph.org Video Test Media [17]. Due to the limited memory of GPU, we cropped the sequences to 176×144 size. In each cropped sequence, input training images are captured from before the in-loop filter during HEVC encoding process (using HM 16.7 software [2]) and the original sequence is used as ground truth images.

For the network implementation, Caffe [18] framework is utilized. We ran 3×10^5 training iterations with ADAM [19] optimizer. The learning rate is adaptively tuned from 1×10^{-3} to 5×10^{-6} until the loss converges.

3. EXPERIMENTAL RESULTS

3.1. Test Environment

In the experiments, the proposed method is tested on test sequences of the JCT-VC common test conditions [20]. Our model is trained only on Y channels of frames, but we applied the model for restoring U, V channels as well. The encoder configuration is set to the 'All Intra - Main' configuration. 4

Table 1: Comparison on various model configurations and prior works on class D testset/Y channel/QP37

Feature Scale	Coding Parameters	Residual Blocks (M)	Model Parameters	PSNR (dB)	Gain (dB)
Single scale	Not used	5	372392	31.071	0.825
Single scale	Not used	15	1109752	31.084	0.837
1 & 1/4 scale	Not used	5	813456	31.151	0.905
1 & 1/4 scale	Not used	15	2288176	31.222	0.975
1 & 1/4 scale	Concat	5	826000	31.226	0.980
1 & 1/4 scale	Concat	15	2300720	31.208	0.961
1 & 1/4 scale	Adaptation net	5	823440	31.233	0.986
1 & 1/4 scale	Adaptation net	15	2298160	**31.303**	**1.056**
Input Image (before in-loop filter)			-	30.247	-
HEVC In-loop Filter [3, 4]			-	30.517	0.270
VDSR [7]			664704	30.972	0.725
VRCNN [1]			54512	31.007	0.760

different models are trained according to QP22, QP27, QP32, and QP37 using training images from corresponding QP settings of HEVC encoder.

3.2. Ablation Studies on the Network

To examine how each component of the proposed model affects the performance, we carried out several controlled experiments with variable components. For this experiment, our approach is evaluated on all sequences in class D [20].

First, we compare the single scale network against the multi-scale network. Table 1 shows that multi-scale network improves the performance by 0.08 - 0.14 dB. On the top of that, the additional coding parameters (CPs) which are CU and TU also enhances the PSNR of the output images. Concatenating CP images along channel axis of an input image is helpful in the 5 residual-block network, but makes worse for the 15 residual-block network. However, a gain of 0.08 dB is obtained through the CP adaptation network (pre-network) for both the 5 and 15 residual-block networks. We can observe that the CP information helps to localize the compression artifacts, but it requires preprocessing network for associating different modalities. The usage of CP in 5 residual-block network has slightly better performance than the 15 residual-block network without CP. So, we can reduce the amount of computation into one-third by sacrificing a small performance gain.

The PSNRs and the number of model parameters of the prior works [1, 7] are also included in Table 1. Although VDSR has a larger number of variables than VRCNN, it fails to find a better optimal point and performance is worse than VRCNN. Our method can alleviate the problem of converging to the local minimum using the structure of the residual network and the batch normalization layers.

Table 2: BD-rate comparison to the HEVC baseline

		Average BD-rate		
Testset	Network	Y	U	V
classC	VDSR	-4.4%	-5.1%	-5.7%
	VRCNN	-4.3%	-6.0%	-6.9%
	MMS-net ($M = 5$)	-7.7%	-9.7%	-11.3%
	MMS-net ($M = 15$)	**-9.3%**	**-13.1%**	**-15.7%**
classD	VDSR	-4.1%	-5.1%	-6.5%
	VRCNN	-3.6%	-5.8%	-7.3%
	MMS-net ($M = 5$)	-6.5%	-8.6%	-10.9%
	MMS-net ($M = 15$)	**-7.7%**	**-10.9%**	**-13.7%**

3.3. Comparison with the State-of-the-art Methods

We also compare MMS-net with the existing state-of-the-art networks in terms of the BD-rate measure. For fair comparison, we trained VDSR [7] and VRCNN [1] on our training dataset until those models converge. Due to the limitation of GPU memory, we only tested the models on the sequences in Class C and D. In Table 2, MMS-net ($M = 15$) achieves the superior performance by reducing average 8.5 % of BD-rate for the Y channel over the HEVC baseline. On the aspect of the model generalization, it is noteworthy that the model is trained on 176×144 images but works well for larger resolutions of images. In addition, this model is trained only on the Y channel, but further reduces the BD-rate on the U and V channels.

For subjective quality comparison, some example images are shown in Figure 1. The results demonstrate that the proposed MMS-net effectively removes the blocking artifacts while preserving major contents and sharp edges of an image.

4. CONCLUSION

In this paper, we proposed a novel multi-modal/multi-scale convolutional neural network (MMS-net) architecture for replacing existing HEVC in-loop filter. MMS-net can successfully remove blocking artifact by exploiting CTU information inside compressed bit stream. Also, the coarse-to-fine approach of the multi-scale network is proven to be advantageous for quantized image restoration. For future work, we will explore ways to utilize other coding information, such as QP, motion vectors, and intra prediction modes, for image restoration.

5. REFERENCES

[1] Y. Dai, D. Liu, and F. Wu, “A convolutional neural network approach for post-processing in HEVC intra coding,” in *Proceedings of the International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[2] “HM reference software: version 16.7,” Available at

https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7/.

[3] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Auwera, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, 2012.

[4] C. M. Fu, E. Alshina, A. Alshin, Y. W. Huang, C. Y. Chen, C. Y. Tsai, C. W. Hsu, S. M. Lei, J. H. Park, and W. J. Han, "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1755–1764, 2012.

[5] "ITU-T recommendation H.265: High efficiency video coding," http://www.itu.int/rec/T-REC-H.265.

[6] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.

[7] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.

[9] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," *arXiv preprint arXiv:1612.02177*, 2016.

[10] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 769–777.

[11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017.

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

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[17] "Xiph.org video test media," Available at https://media.xiph.org/video/derf/.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, Jo. 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*. ACM, 2014, pp. 675–678.

[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference for Learning Representations*, 2014.

[20] F. Bossen, "Common test conditions and software reference configurations," Tech. Rep. JCTVC-H1100, Geneva, CH, January 2013.

[21] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 576–584.