Compression Artifact Removal with Ensemble Learning of Neural Networks

Yueyu Hu[1†] Haichuan Ma[2†] Dong Liu[2] Jiaying Liu[1]

[1]Peking University, Beijing, China
[2]University of Science and Technology of China, Hefei, China

Abstract

We propose to improve the reconstruction quality of DLVC intra coding based on an ensemble of deep restoration neural networks. Different ways are proposed to generate diversity models, and based on these models, the behavior of different integration methods for model ensemble is explored. The experimental results show that model ensemble can bring additional performance gains to post-processing on the basis that deep neural networks have shown great performance improvements. Besides, we observe that both averaging and selection approaches for model ensemble can bring performance gains, and they can be used in combination to pursue better results.

1. Introduction

The amount of image/video data has grown rapidly in the past decade, which brings great challenges to both transmission and storage. To meet these requirements, most of the existing image/video coding schemes perform lossy compression. However, the quantization process in the lossy compression pipeline causes loss of information, leading to artifacts such as blocking, ringing and blurring. As a response to these artifacts, post-processing has been proposed in video compression standards, such as Deblocking Filters (DF) and Sample Adaptive Offset (SAO) in HEVC [10]. In recent years, witnessing the success of deep learning in computer vision tasks, such as super-resolution [3, 5] and denoising [14, 11], researchers have tried to employ deep learning tools to perform post-processing, and have achieved remarkable progress [2, 14, 11, 1, 9, 4, 6].

Since the introduction of deep neural networks into post-processing, it's performance has gradually increased due to the art of designing networks, just like other computer vision tasks. However, what is different in the compression artifact reduction is that specific information can be transmitted from encoder to decoder. Taking advantage of this feature, some in-loop filtering studies propose to determine whether to use a post-processing neural network by the encoder, according to rate-distortion (RD) performance, and transmit flags to decoder [9, 16]. These methods can be viewed as choosing between two neural networks, one of which corresponds to the identity function, i.e. $f(x) = x$. Further developing this idea, multiple post-processing networks are proposed to use to further improve the RD performance of codec [4, 6].

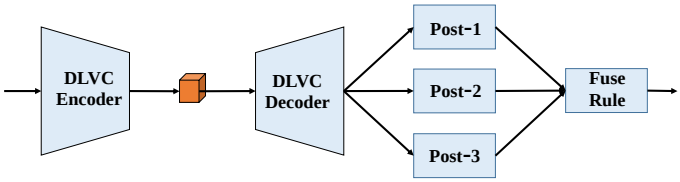

Figure 1: The post-processing framework with ensemble of different neural networks, in which three post-processing models are depicted.

Generally speaking, [9, 16, 4, 6] are typical ensemble manners by integrating multiple models to improve the overall prediction performance. In the area of ensemble learning, integration is used to describe the method of fusing multiple models, which often includes averaging, voting, etc. Selection, which is employed by the above works [9, 16, 4, 6], is usually not the common approach since it is unable to obtain ground truth in most common cases. Without considering the bit rate, the selection is undoubtedly the ideal approach for integration in post-processing. However, since compression is always the trade-off between the rate and distortion, selection may not be always the best under the condition of limited bandwidth. What can other integration methods, such as averaging integration, bring to the ensemble of neural networks in the post-processing task?

To answer this question, we make various attempts to generate diverse neural networks, and mainly demonstrate the behavior of averaging integration method in the ensemble of post-processing neural networks for improving the reconstruction quality of DLVC [7] intra coding. We also

[†]Equal contribution.

conduct experiments to compare it with the selection integration, to better show the differences between the two approaches. At the same time, we explore the possibility of combining the two integration manners, which will bring different inspirations for model ensemble in the post-processing area.

2. Method

Both the basic compression algorithm and the post-processing affect the compression performance. Choosing a good codec is always the first step for pursuing a high compression ratio. In this section, we first introduce the used codec, DLVC, and then introduce the architecture of post-processing neural networks for ensemble learning.

2.1. DLVC

DLVC is developed as a proposal in response to the joint call for proposals (JCfP) on video compression with capability beyond HEVC. It features deep learning-driven coding tools, *i.e.* CNN-based in-loop filter (CNNLF) and CNN-based block adaptive resolution coding (CNN-BARC), both of which are based on CNN models [7]. In the experiments of this paper, we disabled two deep tools of DLVC, which makes it have similar compression performance compared with VTM-7.0*, the reference software of the upcoming H.266/VVC†.

2.2. Post Processing Neural Network

As shown in Fig. 1, the neural networks will be used in the way of model ensemble for post-processing reconstructed images. Different fuse methods will be discussed in the following chapters, mainly including averaging and selection. Now we first briefly introduce the involved neural networks.

RCAN. The model is a very deep residual channel attention neural network, which has been proven to be an excellent super-resolution model [15]. We removed its up-scale layer and used it to post-process the reconstructed images of DLVC. To show its origin, we still call it RCAN.

PRN+. We also adopt the Progressive Rethinking Network (PRN) [13] to construct the post-processing model. The network is originally designed for in-loop filters in video codecs. The Progressive Rethinking Block (PRB) brings in improved capacity for the network. Based on the architecture, we further improve the capacity by deepening the network to twice the original depth. We also change the input and output dimension to adapt to the post-processing task on images of the RGB color space. The improved network is called PRN+ in the experimental analysis.

*https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tags/VTM-7.0

†At the time of writing this paper, H.266/VVC is not officially published yet.

2.3. Ensemble and Signalling

Two ensemble methods are utilized in the construction of the codec. For the signal-free ensemble, we conduct a pixel-wise average over the images produced by different models. Therefore, no additional bits are used to signal the flags along with the bit-stream. For the other ensemble method, we partition the images into blocks and we select one block between those produced by the two models. A flag is signaled to the bit-stream to indicate the selection result. The flags for all the blocks are raster-scanned and we utilize Context Adaptive Binary Arithmetic Coding (CABAC) [8] to encode the flags with its entropy, and the bit-stream is concatenated with the main bit-stream produced by DLVC.

3. Experimental Results

3.1. Experimental Setting

Two image sets, *i.e.* CLIC training set and DIV2K [12], are utilized to train the proposed post-processing models. The models take the reconstructed images as the input and are trained to produce the original images. To generate the training data, the images are first transformed from RGB color space to YUV 4:2:0, the input color format of DLVC, before they are lossily compressed by DLVC. The reconstructed images in YUV 4:2:0 are transformed back to RGB color space, which forms the input data for network training. All resampling is conducted using bilinear interpolation. Different interpolation methods show little differences in PSNR for the resulting images in RGB color space.

The evaluations are conducted on the CLIC 2020 validation set. The images are converted to YUV 4:2:0 in the same way as the training data. We first encode the images in the validation set with QP in the range $[36, 38]$ using DLVC, without the built-in CNNLF as a baseline method. With the reconstructed images and the corresponding bit-streams, we select a set of images that can be encoded into bit-streams with the average bit-per-pixel less than 0.15, while the MSE on this set is minimized. The following experimental results are reported on this testing condition.

3.2. Post Processing

In this section, we evaluate the improvement in the quality of reconstructed images brought by the enhancing post-processing models. Note that DLVC implements CNNLF tool to utilize neural networks for in-loop filtering. We post-process the reconstructed images and we evaluate PSNR and MS-SSIM on the processed images in RGB color space. We first calculate a sum of squared error for all the images in the dataset, and then we calculate an overall mean of the squared error, with which we get the PSNR. MS-SSIM is evaluated for all the images and then averaged across the dataset, weighted by the number of pixels for each image.

Table 1: Comparison of single-model post-processing methods, evaluated on PSNR and MS-SSIM.

Methods	PSNR (dB)	MS-SSIM
DLVC	31.9957	0.9596
Built-in	32.1361	0.9612
PRN+	32.3955	0.9620
RCAN	32.4508	0.9624

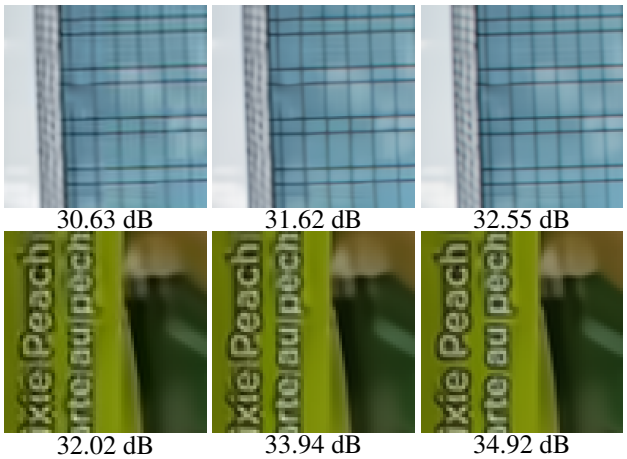

Figure 2: Patches of post-processed images.

The results of single-model methods for enhancing post-processing are illustrated in Table 1, where we compare the performance of the *built-in* in-loop filter in DLVC and two network architectures, *i.e.* PRN+ and RCAN, which are trained on the same dataset.

As shown, upon DLVC with Deblocking Filters (DF) and Sample Adaptive Offset (SAO), both the built-in version and the out-loop versions, achieve improvements in quality on the decoded images. More sophisticated networks and training techniques can bring in extra improvements. A comparison of visual quality is provided in Fig. 2. As illustrated, ringing artifacts exist in DLVC reconstructed images. Some parts of the sharp edges are even missing due to the loss of high-frequency components. Post-processing methods reduce such artifacts, while models achieving higher quantitative quality provide sharper edges and tend to recover the missing parts.

3.3. Empirical Study on Model Ensemble

Model ensemble has been widely utilized in machine learning-driven tasks to improve performance. In this section, we explore the possibility of combining models of different architectures, training image sets, QPs to generate the training set, and those trained with differently screened data.

Table 2: Evaluation of quantitative quality, with combinations of PRN+ models trained using images encoded with different QPs on datasets with mixed CLIC and DIV2K images.

Setting	PSNR (dB)	MS-SSIM
QP32	32.2853	0.9614
QP37	32.3855	0.9620
QP42	32.2626	0.9615
32+37	32.3654	0.9618
42+37	32.3653	0.9619
32+42	32.3691	0.9618
32+37+42	**32.3939**	**0.9621**

3.3.1 QP-Driven Ensemble

We first evaluate the performance by combining models trained on images compressed using different values of QPs. We encode the training images with QPs in $\{32, 37, 42\}$, respectively. These images are used to train the same network architecture. The evaluation results of quantitative quality on the selected images corresponding to Table 1 are shown in Table 2. It shows that models trained with QPs of ranges different from the validation set achieve lower quantitative quality, and averaging pixel values on the images for any two of the three models does not produce a better performance than a single model of the appropriate QP. However, the average ensemble of all three models can result in improved performance over any single-model or two-model averaging settings. The results indicate that models trained with inappropriate QPs have biases. The bias can result in degraded quality if the model is averaged with another model trained with a more appropriate training setting. Despite that, the bias can be largely reduced by averaging two models with potentially different biases.

3.3.2 MSE-Driven Ensemble

Another way to generate different models for the ensemble is to train the models with different subsets of the original training data. In this experiment, we split the training set by calculating patch-wise MSE between the images encoded using QP 37 and the original ones. The patches are divided into three subsets, corresponding to three levels of MSE, *i.e* high, median and low. The experiment is conducted with RCAN models. The results are shown in Table 3. Splitting the dataset according to MSE also introduces biases in model training, while we observe that patches of higher MSE, corresponding to tougher cases, result in higher performance. The results also indicate that a particularly biased model can make bad effects on the averaged performance.

3.3.3 Image-Set-Driven Ensemble

Training images is another factor to influence model performance. We conduct the experiments on the PRN+ archi-

Table 3: Evaluation of quantitative quality, with combinations of RCAN models trained with different subests of the training set, divided by MSE. *Full* refers to the original dataset without splitting. *A+B* refers to averaging the outputs of models from *Low*, *Median* and *High* and *All* refers to averaging all four models.

Setting	PSNR (dB)	MS-SSIM
Low	32.2863	0.9617
Median	32.4018	0.9623
High	32.4237	0.9622
Full	32.4508	0.9624
L+F	32.4252	0.9623
M+F	32.4585	**0.9625**
H+F	32.4604	0.9624
L+M+H	32.4564	**0.9625**
H+M+F	**32.4757**	**0.9625**
All	32.4701	**0.9625**

Table 4: Evaluation of quantitative quality, with combinations of PRN+ models trained with different image sets.

Setting	PSNR (dB)	MS-SSIM
Mixed	32.3855	0.9620
CLIC	32.3955	0.9620
DIV2K	32.3806	0.9620
M+C	32.3931	0.9620
M+D	32.3893	0.9620
C+D	32.3949	0.9620
All	**32.3987**	**0.9621**

tecture, on two different datasets, *i.e.* CLIC and DIV2K. We first train a model with the mixed dataset and we finetune the model to produce two different models, each tuned on either CLIC or DIV2K dataset. The results are shown in Table 4. The choice of datasets affects the performance, where the CLIC dataset is shown to better fit the validation set. Averaging outputs show little improvements in performance.

3.3.4 Ensemble of Different Architectures

We conduct a cross-architecture ensemble for PRN+ and RCAN models. The results are shown in Table 5. As shown, the cross-model ensemble improves the overall quantitative quality, and averaging can also be applied to two sets of already ensemble models that have different architectures. We then compare the result of averaging pixels to block-wise selection. Block size 96×96 is chosen to not exceed the limit of 0.15 bpp among the images. It is observed that for outputs that have not been ensemble, averaging the pixels achieves higher quality without consuming any bit-rate. However, for ensemble outputs, block-level selection may have the potential to bring in more improvements.

We conduct a further investigation into the block-wise

Table 5: Evaluation with combinations of PRN+ and RCAN models. For two model A and B, *A+B* refers to averaging pixels while *A/B (N)* means conducting block-level selection with the block size *N*. *P* and *R* stand for PRN+ and RCAN, respectively, while *P-All* and *R-HMF* correspond to the best ensemble results in Table 4 and Table 3.

Setting	PSNR (dB)	MS-SSIM
PRN+	32.3955	0.962
RCAN	32.4508	0.9624
P+R	32.4651	0.9624
P/R (96)	32.4646	0.9625
P-All + R-HMF	32.4783	0.9625
P-All / R-HMF (96)	32.4842	0.9626

Table 6: Evaluation of block-level ensemble of different models. *P* and *R* stand for PRN+ and RCAN, respectively, while *P-All* and *R-HMF* correspond to the best results in Table 4 and Table 3. *Ratio* show the ratio of selected patches that originally comes from the first model. *bpp+* indicate the increase in bpp to signal the flags.

Setting	Patch	Ratio	PSNR (dB)	bpp+
P/R	96	30.55%	32.4646	9.6e-5
R-M/R-HMF	96	28.04%	32.4794	9.4e-5
P-All/R-HMF	96	23.29%	32.4841	8.4e-5
P-All/R-HMF	128	19.67%	32.4822	4.3e-5
P-All/R-HMF	64	28.08%	32.4884	2.1e-4
P-All/R-HMF	48	31.18%	32.4928	3.8e-4

selection method. As shown in Table 6, block-wise ensemble with streamed flags can further improve quantitative quality, with only a slight amount of increase in bit-rate. Surely the finer partitioning of the images produces higher results while consuming more bits for the flags. It is also observed that models showing some distinctions may achieve a better result with the ensemble. As illustrated in Table 5, two similar models (*R-M* and *R-HMF*), though with higher averaged PSNR, do not outperform two distinct models (*P-All* and *R-HMF*) when conducting the block-wise ensemble. To design a collaborative training technique that considers rate-distortion optimization might be a promising direction for future research in learned image compression.

4. Conclusion

In this paper, we explored the possibility of combining multiple models in the post-processing task. These models are generated in different ways, including designing different network architectures and training with different image sets. We also compared two different integration methods, averaging and selection, and found that both the two methods can bring performance gains, and they can be superimposed for better results.

References

[1] Yuanying Dai, Dong Liu, and Feng Wu. A convolutional neural network approach for post-processing in HEVC intra coding. In *Proc. of International Conference on Multimedia Modeling*, 2017.

[2] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proc. of International Conference on Computer Vision*, 2015.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[4] Chuanmin Jia, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Jiaying Liu, Shiliang Pu, and Siwei Ma. Content-aware convolutional neural network for in-loop filtering in high efficiency video coding. *IEEE Transactions on Image Processing*, 28(7):3343–3356, 2019.

[5] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[6] Weiyao Lin, Xiaoyi He, Xintong Han, Dong Liu, John See, Junni Zou, Hongkai Xiong, and Feng Wu. Partition-aware adaptive switching neural networks for post-processing in HEVC. *IEEE Transactions on Multimedia*, 2019.

[7] Dong Liu, Yue Li, Jianping Lin, Houqiang Li, and Feng Wu. Deep learning-based video coding: A review and a case study. *ACM Computing Surveys*, 53(1):1–35, 2020.

[8] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand. Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):620–636, 2003.

[9] Xiandong Meng, Chen Chen, Shuyuan Zhu, and Bing Zeng. A new HEVC in-loop filter based on multi-channel long-short-term dependency residual networks. In *Proc. of Data Compression Conference*, pages 187–196. IEEE, 2018.

[10] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[11] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. MemNet: A persistent memory network for image restoration. In *Proc. of International Conference on Computer Vision*, 2017.

[12] Radu Timofte, Shuhang Gu, Jiqing Wu, Luc Van Gool, Lei Zhang, Ming-Hsuan Yang, Muhammad Haris, et al. NTIRE 2018 challenge on single image super-resolution: Methods and results. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018.

[13] Dezhao Wang, Sifeng Xia, Wenhan Yang, Yueyu Hu, and Jiaying Liu. Partition tree guided progressive rethinking network for in-loop filtering of HEVC. In *Proc. of IEEE International Conference on Image Processing*, 2019.

[14] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[15] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proc. of European Conference on Computer Vision*, 2018.

[16] Yongbing Zhang, Tao Shen, Xiangyang Ji, Yun Zhang, Ruiqin Xiong, and Qionghai Dai. Residual highway convolutional neural networks for in-loop filtering in HEVC. *IEEE Transactions on image processing*, 27(8):3827–3841, 2018.