A Neural-network Enhanced Video Coding Framework beyond VVC

Junru Li, Yue Li, Chaoyi Lin, Kai Zhang, Li Zhang
Multimedia Lab, Bytedance Inc., San Diego CA. 92122 USA
{lijunru, yue.li, linchaoyi.cy, zhangkai.video, lizhang.idm}@bytedance.com

Abstract

This paper presents a hybrid video compression framework, aiming at providing a demonstration of applying deep learning-based approaches beyond conventional coding framework. The proposed hybrid framework is established over the Enhanced Compression Model (ECM) of which the core is the Versatile Video Coding (VVC) standard. We propose to integrate a series of enhanced coding tools, such as block partitioning, intra prediction, and inter prediction to further remove the spatial and temporal redundancy. Moreover, deep learning-based technologies including loop filter and super resolution are involved to restore the compression distortion. Compared with the VVC software VTM-11.0, experimental results demonstrate the effectiveness of the proposed learning-based framework, leading to 25.81%, 35.08%, and 37.54% bit-rate savings for Y, Cb and Cr components, respectively under random access configuration. In addition, the proposed framework achieves 39.313 and 32.050 PSNRs in the test set under 1 Mbps and 0.1 Mbps video compression tracks of CLIC-2022. 33.522, 30.758, and 28.300 in terms of PSNR are obtained in 0.3 bpp, 0.15 bpp, and 0.075 bpp image compression tracks.

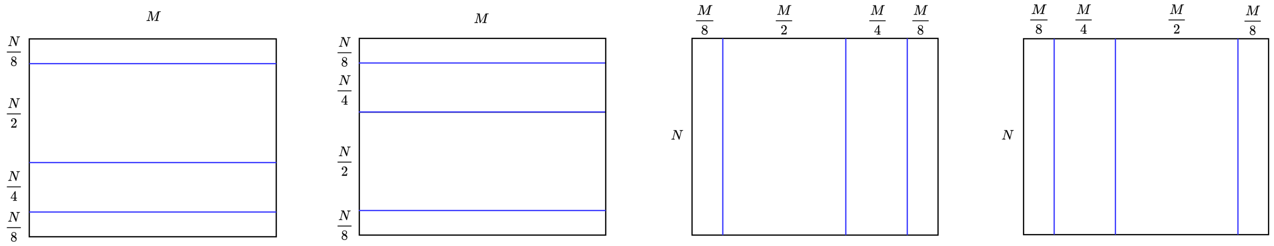
1. Introduction

Recent years have witnessed the rapid development of the video-oriented applications. The video data volume increases rapidly, which constantly brings new challenges to image and video coding. Video coding technologies have been explored for several decades, which aims at more compactly representing the visual signals with tolerable perceivable distortions, thus facilitating the storage and transmission. Currently, the mainstream video coding standards such as the High Efficiency Video Coding (HEVC) [16], the Versatile Video Coding (VVC) [4], and the third generation of Audio and Video coding Standard (AVS3) [20] are deployed with the classical hybrid coding framework. The prediction coding [5–9], transformation and quantization coding [15, 24], entropy coding [17], and in-loop filtering [18] are delicately cooperated, such that the spatial, temporal, statistical redundancies could be effectively removed.

Due to the limited capacity of the transmission bandwidth and storage spaces, video coding technologies evolve rapidly. The next generations of video coding standards in terms of the VVC and AVS3 were finalized in 2020, emphasizing versatility and adaptability of the video codec in various application scenarios. Almost all modules are enhanced in the VVC and AVS3. In particular, VVC and AVS3 both adopt more complicated coding unit (CU) partitioning structure, wherein the quad-tree (QT) nested binary-tree and ternary-tree/extended quad-tree partitioning [19] is employed to better adapt multiplex video contents. Moreover, the number of angular intra prediction is doubled [7], in order to capture the arbitrary edge directions more efficiently. Cross-component prediction [10] is considered as additional chroma intra prediction mode such that the redundant information existing in different color components could be eliminated. Affine motion compensation [21] is adopted by the VVC and AVS3 to cope with the non-translation motions.

Deep-learning based coding tools have attracted many attentions in the exploration experiments, which mainly concentrate on modifying the prediction and filtering modules. The promising deep-learning based coding tools bring significant performance improvement in terms of the compression performance and reconstruction quality, exhibiting remarkable restoration and non-linear modeling capability. Due to the limited computing resources at the user-end, the high-complexity decoder induced by the deep learning module is still unacceptable, which hinders the further standardization of the deep-learning based coding tools.

In this paper, we propose a learning-based video coding framework, which successfully harmonizes the traditional coding tools and deep-learning based coding tools, leading to significant improvement of the compression performance. To be more specific, the basic framework is constructed upon the Enhanced Compression Model 3.1 (ECM) [13], cooperating with more advanced CU partitioning structure, enhanced prediction tool, learning based fil-



(a) UQT-H1. (b) UQT-H2. (c) UQT-V1. (d) UQT-V2.

Figure 1. Illustration of the UQT partitioning.

tering and super resolution. The proposed framework significantly surpasses the VTM-11.0 [14] with 28.28% BD-Rate [2] gain. Moreover, the hybrid framework is expected to promote the future research and development of the traditional and learning-based video compression.

2. Framework

In this section, we elaborate the components of the proposed video compression framework which is built on top of the ECM. A series of efficient conventional coding tools including block partitioning, intra prediction, inter prediction are involved. Moreover, neural-network based video coding technologies are adopted to nest with the conventional hybrid coding framework. The proposed learning-based framework achieves the high compression performance owing to the cooperation of the advanced coding technologies.

2.1. Unsymmetric Quaternary Tree

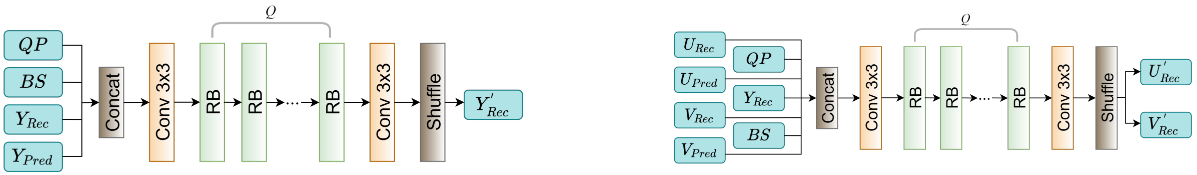
Coding unit partitioning has been a long-standing problem in the block-based hybrid video coding, which determines the shape and scales of the basic coding unit. Flexible partitioning framework plays crucial roles in depicting the diverse local contents.

In this paper, we introduce an unsymmetric quaternary tree (UQT) partitioning structure, with the goal of improving the coding efficiency for larger blocks [22]. As shown in Fig. 1, there are four types of UQT partitioning structures. According to the location of the largest sub-block, the splitting shape of UQT could be noted as left, right, top and bottom, corresponding to UQT-V1, UQT-V2, UQT-H1, and UQT-H2. Unlike the QT, UQT divides a block into four sub-parts asymmetrically along one certain direction. For the horizontal direction, UQT-H1 and UQT-H2 split the $M \times N$ block into one $M \times N/2$, one $M \times N/4$ and two $M \times N/8$ sub-parts. UQT-V1 and UQT-V2 are along with the vertical direction and divide the $M \times N$ block into one $M/2 \times N$, one $M/4 \times N$ and two $M/8 \times N$ sub-parts. Compared with the QT and Multi-Type Tree (MTT) [4], the smallest sub-blocks generated by UQT could achieve deeper depth with once splitting and capture the rich details more effectively. Moreover, UQT produces a new partitioning pattern which could not be achieved by QT or MTT with identical splitting times. It is worthy of mentioning that the dimensions of sub-blocks are limited in the power of two, such that it is unnecessary to involve new transform shapes.

2.2. History-based Affine Model Inheritance

There has been a consensus regarding the inter prediction technique in video-based compression that the temporal redundancy can be efficiently removed by the motion compensation. The motion model with certain motion parameters forms the basic skeleton of motion compensation. Translational motion compensation model depicts the rigid motion in videos, which assumes that the motion objects belongs to translation movement. Affine motion compensation model is employed to capture the complex motion scenes, such as rotation and zooming.

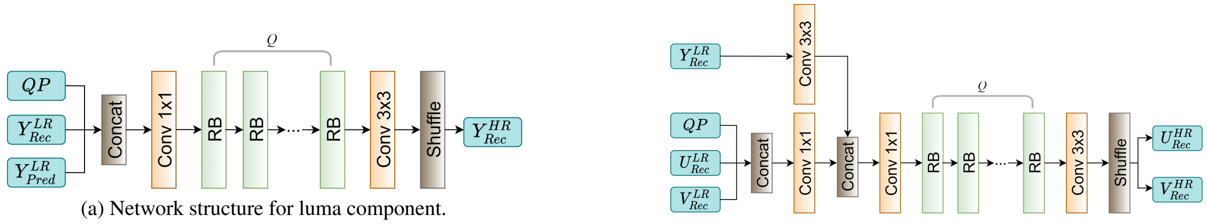
In VVC, history-based motion vector prediction (HMVP) has been adopted [23]. History-based affine model inheritance (HAMI) is integrated into the proposed framework to reduce the long-term correlation of model parameters among the coding units. Inspired by HMVP, HAMI fully explores the history model parameters of previous coded blocks with affine mode. More specifically, a history-parameter table (HPT) which records sets of affine parameters, is elaborately maintained with limited capacity. For each category indicated by reference picture list and reference index, at most two entries can be held in HPT. In particular, affine parameters of coded blocks are grouped into a candidate to update the HPT on-the-fly similar to the HMVP after encoding or decoding an affine-coded block. To take full advantage of HAMI, the parameter candidate in HPT can be utilized to derive the motion vector (MV) of the current block with the base MV from the neighboring blocks. With HAMI method, we could increase the candidates of the affine AMVP, affine merge and regular merge modes to decrease the redundancies of motion parameters.



(a) Network structure for luma component.

(b) Network structure for chroma components.

Figure 2. Illustration of the network structure regarding the proposed CNN-based in-loop filter.



(a) Network structure for luma component.

(b) Network structure for chroma components.

Figure 3. Illustration of the network structure regarding the proposed CNN-based super resolution.

2.3. CNN-based In-Loop Filter

In-loop filters are adopted for coding artifacts removal in hybrid video coding, such as deblocking filter, sample adaptive offset and adaptive loop filter. The differences between the original input and the reconstruction could be mitigated with in-loop filters. The surging of the convolutional neural network (CNN) impels the further exploration of the CNN based in-loop filters, which is anticipated to further enhance the qualilty of the reconstructions.

As shown in Fig. 2, a CNN-based filtering method with adaptive parameter selection is employed [11]. For guiding the enhancement, a series of intermediate compression information serving as extra side information conducts as the input of the network, which could supplement the prior knowledge. Specifically, prediction signals, boundary strength generated during the compression process, as well as the quantization parameter (QP) are employed as the auxiliary information of network. Moreover, regarding the filter model of intra slice, the partitioning information is involved as the additional input. With the involvement of the QP as the network input, the model is unified to adapt to various quality levels. Adaptive selection mechanism is involved to adapt the contents at slice and block levels wherein the model usage could be indicated by the signaled flags.

2.4. CNN-based Super Resolution

Resampling is a fundamental strategy frequently used in image and video compression. With a serial of operations regarding down-sampling and up-sampling, the coding efficiency could be improved, especially in terms of high resolution frame with more compact representation inside the texture. VVC has supported the reference picture resampling (RPR) which involves the adaptive resolution variation in hierarchical reference structure. Owing to the power of neural network, the up-sampling in RPR could be further elevated.

Herein, we integrate a CNN-based super resolution method which leverages the side information to perceive compression distortion during the encoding or decoding process [12]. More specifically, the proposed method involves the prediction signals as the auxiliary information of reconstruction signal where the prediction signals could provide a guideline of characteristics such as the textural features and directional features. To handle the various quality levels with only one single model, QP map under sequence level is fed into the network. In the network part, those side information concatenated with reconstruction are fed into a convolutional layer and the output is then followed by several residual blocks and a convolutional layer. Finally, the high-resolution reconstruction is generated by a shuffle layer. Considering the diversity of different components, the super resolution models for luma and chroma components are designed and shown in Fig. 3, respectively. In particular, the luma reconstruction located at the temporal collocated position of chroma component is exploited to guide the chroma super resolution based on the texture extracted from luma component. It is worth mentioned that the proposed CNN-based super resolution is more effective for resampling in the low bit-rate scene.

3. Experimental Results

3.1. Performance with JVET Data Set

Herein, we conduct the experiments to compare the proposed framework with the VVC where the VTM-11.0 [14]

Table 1. Performance of the proposed framework under RA configuration compared with the VTM-11.0

Class	Sequence	BD-Rate [%]			
		YCbCr	Y	Cb	Cr
A1	Tango2	-31.71	-28.41	-40.87	-36.39
	FoodMarket4	-26.79	-24.80	-31.29	-32.62
	Campfire	-25.37	-23.35	-18.03	-45.29
A2	CatRobot1	-29.95	-27.76	-38.93	-36.11
	DaylightRoad2	-38.67	-34.09	-46.40	-51.68
	ParkRunning3	-21.51	-20.61	-22.48	-29.93
B	MarketPlace	-28.02	-24.19	-42.64	-42.59
	RitualDance	-26.29	-24.46	-32.72	-34.77
	Cactus	-24.27	-22.26	-32.70	-31.45
	BasketballDrive	-29.77	-27.63	-37.10	-35.45
	BQTerrace	-31.10	-25.25	-48.93	-48.81
C	BasketballDrill	-28.99	-27.65	-33.56	-34.58
	BQMall	-28.15	-25.94	-36.42	-36.02
	PartyScene	-28.89	-28.17	-32.69	-30.94
	RaceHorses	-24.79	-22.60	-31.37	-36.44
D	BasketballPass	-27.13	-25.42	-35.33	-33.21
	BQSquare	-38.97	-37.92	-39.30	-46.23
	BlowingBubbles	-25.28	-23.91	-30.63	-29.99
	RaceHorses	-26.48	-24.59	-33.96	-34.97
F	BasketballDrillText	-26.56	-25.65	-30.06	-29.64
	ArenaOfValor	-21.27	-19.88	-26.96	-25.37
	SlideEditing	-13.75	-12.70	-17.48	-17.91
	SlideShow	-23.32	-20.82	-32.81	-31.72
Average(A1)		-27.96	-25.52	-30.07	-38.10
Average(A2)		-30.04	-27.49	-35.94	-39.24
Average(B)		-27.89	-24.76	-38.82	-38.62
Average(C)		-27.71	-26.09	-33.51	-34.49
Average(D)		-29.47	-27.96	-34.80	-36.10
Average(F)		-21.23	-19.76	-26.83	-26.16
Average(A1,A2,B,C)		-28.28	-25.81	-35.08	-37.54

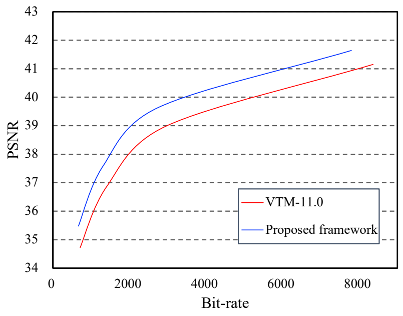


Figure 4. Rate and distortion curves of VTM-11.0 and the proposed framework.

Table 2. Performance of the proposed framework under image and video compression tracks in the test set of CLIC-2022.

Metric	Image Track			Video Track	
Bitrate	0.075 bpp	0.15 bpp	0.3 bpp	0.1 Mbps	1 Mbps
PSNR	28.300 dB	30.758 dB	33.522 dB	32.050 dB	39.313 dB

Table 3. Performance of the proposed framework under image and video compression tracks in the validation set of CLIC-2022.

Metric	Image Track			Video Track	
Bitrate	0.075 bpp	0.15 bpp	0.3 bpp	0.1 Mbps	1 Mbps
PSNR	29.359 dB	31.728 dB	34.287 dB	31.218 dB	37.850 dB

reference software is adopted. In the evaluation process, random access (RA) configuration conforming to the common test condition [3] is used in the experiments. Sequences recommended by JVET is involved in the simulation, including classes A1, A2, B, C, D, and F. The QPs are set as 22, 27, 32, and 37. The coding performance is measured by BD-Rate [2] where negative BD-Rate indicates the performance improvement. The YCbCr BD-Rate is calculated by averaging the PSNR of Y, Cb, and Cr components with weights 6:1:1. Tab. 1 shows the coding performance of the proposed hybrid architecture compared with the VTM-11.0 for each sequences under RA configuration. It can be observed that the proposed framework achieves 28.28%, 25.81%, 35.08% and 37.54% BD-Rate savings for YCbCr, Y, Cb, and Cr components, respectively on average of class A1, A2, B, and C. The rate and distortion curves of VTM-11.0 and the proposed framework are shown in Fig. 4.

3.2. Performance with CLIC Data Set

In this section, we evaluate the performance conforming to the test conditions [1] recommended by the CLIC-2022 challenge. In the video and image compression tracks, 30 videos and images are involved, respectively. The coding performance is measured by the averaged PSNR per pixel under the limited bit-rates among all sequences where the higher PSNR demonstrates the better performance. The performance in the test set is shown in Tab. 2. The proposed framework achieves 39.313 and 32.050 dB in terms of PSNR under 1 and 0.1 mega bits per second (Mbps) of video tracks. Under 0.3, 0.15, and 0.075 bits per pixel (bpp) of image tracks, the associated PSNR values are 33.522, 30.758, and 28.300 dB. As show in Tab. 3, we also provide the experimental results in the validation set. In video tracks, the PSNR values of the proposed framework are 37.850 and 31.218 dB. In addition, 34.287, 31.728, and 29.359 dB in terms of PSNR are acquired in image tracks.

4. Conclusion

In this paper, an artificial-intelligence-based video compression framework is proposed. A series of enhanced compression tools are involved in the framework, and meanwhile the CNN-based in-loop filtering and super resolution are cooperated. The proposed framework significantly surpasses the VTM with 28.28% BD-Rate gain. Moreover, the proposed framework achieves remarkable compression quality for image and video compress tracks in CLIC-2022.

References

[1] CLIC 2022. The 5th workshop and challenge on learned image compression. https://compression.cc/tasks. 4

[2] G. Bjontegaard. Calculation of average PSNR differences between rd-curves. *ITU-T SG.16 Q.6 VCEG-M33*, 2001. 2, 4

[3] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Shring. VTM common test conditions and software reference configurations for SDR video. *Joint Video Exploration Team (JVET), doc. JVET-T2010*, Oct. 2020. 4

[4] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan, and Y. K. Wang. Developments in international video coding standardization after avc, with an overview of versatile video coding (VVC). *Proceedings of the IEEE*, pages 1–31, 2021. 1, 2

[5] B. Bross, H. Schwarz, D. Marpe, and Thomas T. Wiegand. CE 3: Multiple reference line intra prediction. *Joint Video Exploration Team (JVET), doc. JVET-K0051*, 2018. 1

[6] B. Bross, H. Schwarz, D. Marpe, and T. Wiegand. CE 3.3 related: Wide angular intra prediction for non-square blocks. *Joint Video Exploration Team (JVET), doc. JVET-L0279*, 2018. 1

[7] N. Choi, Y. Piao, K. Choi, and C. Kim. CE 3.3 related: Intra 67 modes coding with 3 MPM. *Joint Video Exploration Team (JVET), doc. JVET-K0529*, 2018. 1

[8] H. Gao, H. Gao, X. Chen, S. Esenlik, J. Chen, and E. Steinbach. Decoder-side motion vector refinement in VVC: Algorithm and hardware implementation considerations. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2020. 1

[9] H. Gao, R. Liao, K. Reuzé, S. Esenlik, E. Alshina, Y. Ye, J. Chen, J. Luo, C. Chen, H. Huang, W. Chien, V. Seregin, and M. Karczewicz. Advanced geometric-based inter prediction for versatile video coding. In *2020 Data Compression Conference (DCC)*, pages 93–102, 2020. 1

[10] J. Li, M. Wang, L. Zhang, S. Wang, K. Zhang, S. Wang, S. Ma, and W. Gao. Sub-sampled cross-component prediction for emerging video coding standards. *IEEE Transactions on Image Processing*, 30:7305–7316, 2021. 1

[11] Y. Li, L. Zhang, and K. Zhang. iDAM: Iteratively trained deep in-loop filter with adaptive model selection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022. 3

[12] C. Lin, Y. Li, K. Zhang, Z. Zhang, and L. Zhang. CNN-based super resolution for video coding using decoded information. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2021. 3

[13] ECM Repository. https://vcgit.hhi.fraunhofer.de/ecm/ECM. 1

[14] VTM Repository. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tree/VTM-11.0. 2, 3

[15] H. Schwarz, T. Nguyen, D. Marpe, and T. Wiegand. Hybrid video coding with trellis-coded quantization. In *Proc. Data Compress. Conf. (DCC)*, pages 182–191, 2019. 1

[16] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. 1

[17] V. Sze and M. Budagavi. High throughput CABAC entropy coding in HEVC. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1778–1791, 2012. 1

[18] C. Tsai, C. Chen, T. Yamakage, I. Chong, Y. Huang, C. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz, and S. Lei. Adaptive loop filtering for video coding. *IEEE Journal of Selected Topics in Signal Processing*, 7:934–945, 12 2013. 1

[19] M. Wang, J. Li, L. Zhang, K. Zhang, H. Liu, S. Wang, S. Kwong, and S. Ma. Extended coding unit partitioning for future video coding. *IEEE Transactions on Image Processing*, 29:2931–2946, 2020. 1

[20] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma, and W. Gao. Recent development of avs video coding standard: Avs3. In *2019 Picture Coding Symposium (PCS)*, pages 1–5, 2019. 1

[21] K. Zhang, Y. Chen, L. Zhang, W. Chien, and M. Karczewicz. An improved framework of affine motion compensation in video coding. *IEEE Transactions on Image Processing*, 28(3):1456–1469, 2019. 1

[22] K. Zhang, L. Zhang, Z. Deng, N. Zhang, and Y. Wang. Advanced block partitioning methods beyond VVC. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022. 2

[23] L. Zhang, K. Zhang, H. Liu, H. Chuang, Y. Wang, J. Xu, P. Zhao, and D. Hong. History-based motion vector prediction in versatile video coding. In *2019 Data Compression Conference (DCC)*, pages 43–52. IEEE, 2019. 2

[24] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin. Joint separable and non-separable transforms for next-generation video coding. *IEEE Transactions on Image Processing*, 27(5):2514–2525, 2018. 1