Deep Video Compression for P-frame in Sub-sampled Color Spaces

Rongqun Lin, Pingping Zhang, Meng Wang, Shiqi Wang, Sam Kwong
Department of Computer Science, City University of Hong Kong
Email: {rqlin3-c, pinzhang6-c, mwang98-c}@my.cityu.edu.hk, {shiqwang, cssamk}@cityu.edu.hk

***Abstract*—In this paper, we propose a deep video compression method for P-frame in sub-sampled color spaces regarding the YUV420, which has been widely adopted in many state-of-art hybrid video compression standards, in an effort to achieve high compression performance. We adopt motion estimation and motion compression to facilitate the inter prediction of the videos with YUV420 color format, shrinking the total data volume of motion information. Moreover, the motion compensation module on YUV420 is cooperated to enhance the quality of the compensated frame with the consideration of the resolution alignment in the sub-sampled color spaces. To explore the cross-component correlation, the residual encoder-decoder is accompanied with two head-branches and color information fusion. Additionally, a weighted loss emphasizing more on the Y component is utilized to enhance the compression efficiency. Experimental results show that the proposed method can realize 19.82% bit rate reductions on average compared to the deep video compression (DVC) method in terms of the combined PSNR and predominant gains on the Y component.**

***Index Terms*—Deep learning, learned video compression, P-frame, sub-sampled color spaces.**

I. Introduction

With the rapid development of the Internet and digital devices, video data volumes have been tremendously increased, bringing substantial challenges to video coding technologies. During the past several decades, the video coding has been significantly evolved based on the hybrid video coding framework, which is typically constituted with the prediction, transform, quantization and entropy coding. The video coding standards, such as the high efficiency video coding (HEVC) standard [1], the audio and video coding standard (AVS) [2], and the versatile video coding (VVC) standard [3], realize considerable breakthroughs regarding the compression performance, flourishing the further development of the video-oriented applications.

The surging of the deep neural networks impels the exploration of the learning based compression schemes [4]. Existing works regarding the learning based compression can be classified into two categories. The first category replaces the conventional coding module with the deep neural network to enhance the representation capability, which still relies on the hybrid coding framework. In particular, deep neural networks are used to create new intra prediction modes [5] [6] [7], such that the prediction accuracy could be improved. Regarding inter prediction, the deep learning is cooperated to improve the motion estimation precision [8] [9]. To further remove the statistical redundancies, the deep learning based entropy coding method [10] has been proposed, leveraging the deep neural network to estimate the probability distribution of residuals, such that the coding performance could be enhanced. Deep learning based loop filtering methods [11] [12] [13] [14] intend to eliminate the compression degradation by learning an enhancement model on the distorted patches. However, existing methods mainly focus on an individual module in the hybrid video coding framework and the visual signal representation pipeline has been insufficiently utilized.

The second category achieves compression with end-to-end learning framework where the recurrent neural network, convolutional neural network, as well as the generalized divisive normalization layer are delicately cooperated, converting the visual signals to the latent-code with non-linear transform. End-to-end based image compression [15] [16] has successively surpasses the traditional image coding, such as the JPEG2000, the HEVC, and the VVC. However, employing the end-to-end learning network to compress the video is still a challenging task, as the coding procedure is more intricate, bestowing spaces for further investigation. Deep Video Compression (DVC) [17] is the first fully end-to-end video coding framework, which imitates the functionality of the modules in traditional hybrid coding framework and replaces all components with deep neural networks. In [18], the motion information and residual information are jointly optimized and compressed, leading to further reductions of the coding bits. A 3D auto-encoder with auto-regressive prior is utilized in [19] to enhance the entropy estimation. In [20], multiple reference frames and motion information are used in the end-to-end framework to produce better prediction of the current frame, leading to more compact residuals between prediction and current frame. In [21], error propagation and content-aware end-to-end video coding framework have been proposed to improve the coding performance by applying rate-distortion criterion to update network weights.

It has been widely recognized that the YUV color space enjoys significant benefits regarding the energy concentration, making it prevalent in displaying, transmission and the traditional video compression. Human visual system is more sensitive to luma component, such that the luma component is provided with finer coding strategies in traditional video coding. In this manner, the chroma component could be downsampled to reduce visual redundancy. However, most of the existing end-to-end compression methods are implemented in the RGB color spaces. Egilmez *et al.* [22] proposed a learning based image compression framework for sub-sampled color spaces, which could not be directly applied to video compression. As such, the compression potentials with the YUV420 color format are still inadequately excavated. Moreover, tra-

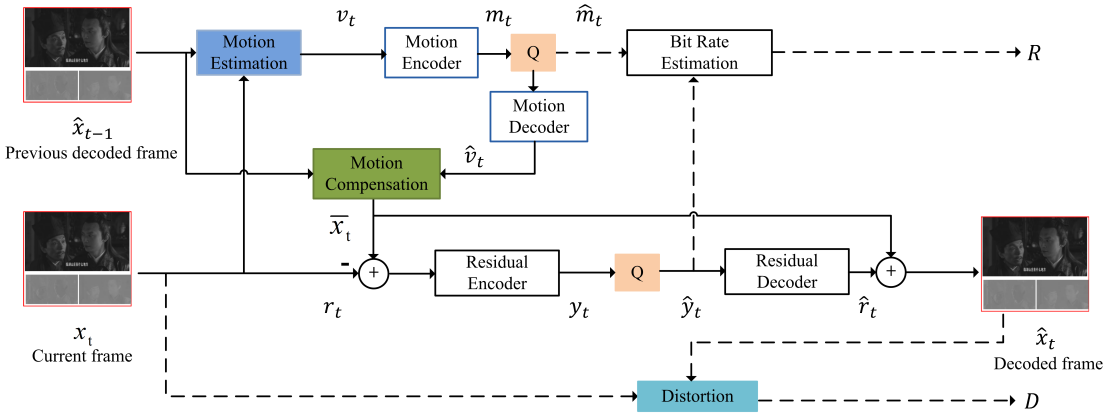

Fig. 1. Illustration of the framework of the proposed method.

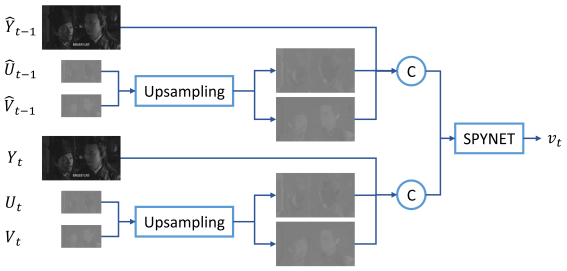

Fig. 2. Illustration of the motion estimation for the videos with YUV420 color format.

ditional video compression standards primarily pursue higher coding performance in sub-sampled color spaces instead of the RGB color spaces in the sense of rate and distortion. The performance gap between learned video compression and the state-of-art traditional video coding standards has not been illustrated sufficiently.

In this paper, we propose an end-to-end video compression method for P-frame in sub-sampled color spaces with the contributions summarized as follows:

- We introduce the motion estimation and motion encoder-decoder on YUV420 to produce compact optical flow for Y, U and V components. Furthermore, we present the motion compensation with resolution alignment and enhancement on each component.
- We apply the residual encoder-decoder with the consideration of cross-component correlation. A proper loss function emphasizing the importance of the Y component is utilized.
- Experimental results demonstrate that the proposed method could improve the previous end-to-end video compression method in sub-sampled color spaces.

II. Proposed Method

A. Framework

The framework of the method in this work is shown in Fig. 1, which follows the hybrid P-frame coding structure in DVC [17]. The method focuses on the sub-sampled color space regarding the Y, U and V with 4:2:0 color format. A series of compression modules are elegantly devised to facilitate the compression of the videos in YUV420 color format. When compressing the current frame x_t, the optical flow v_t between the current frame and the previous decoded frame $\hat{x}_{t-1}$ is firstly calculated through a motion estimation module. Subsequently, the v_t is compressed through the motion encoder. Then, inter frame prediction is conducted for the current frame by referring the reconstructed optical flow $\hat{v}_t$ and the reference frame through the motion compensation module, generating the predicted frame $\bar{x}_t$. The residual r_t between the current frame and predicted frames is calculated and passed over to the residual codec, wherein the residual encoder transforms the r_t to the residual latent code y_t. The bit rate of the quantized residual latent code $\hat{y}_t$ is estimated. At the last stage, the reconstructed residuals $\hat{r}_t$ and the prediction signal $\bar{x}_t$ are combined to generate the reconstruction $\hat{x}_t$ of the current frame, which could serve as the reference frame for the subsequent coding frame.

B. Motion Estimation and Compression with YUV420

Motion estimation is built upon the the SPYNET [23] to generate the optical flow for the video with YUV420 color format. The Y, U and V components of the current input frame x_t are denoted as Y_t, U_t and V_t, respectively. Moreover, the Y, U and V components of the associated reference frame are denoted as $\hat{Y}_{t-1}$, $\hat{U}_{t-1}$, $\hat{V}_{t-1}$, respectively. As shown in Fig. 2, U_t, V_t and U_{t-1}, V_{t-1} are upsampled to align to the resolution of the luma channel. SPYNET computes the optical flow v_t between the concatenation of Y_t, U_t, V_t and the concatenation of $\hat{Y}_{t-1}$, $\hat{U}_{t-1}$, $\hat{V}_{t-1}$. The optical flow v_t, which is with the same resolution as the luma channel, is fed to motion encoder. The motion encoder is composed of convolutional layers and generalized divisive normalization (GDN) layers, as illustrated in Fig. 3. The motion decoder consists with deconvolutional layers and inverse GDN layers, synthesising the quantized latent code to reconstruct the optical flow $\hat{v}_t$.

C. Motion Compensation on YUV420

Motion compensation aims at enhancing the quality of the warping frame. More specifically, the warping frame is generated from the reconstructed optical flow $\hat{v}_t$ and the

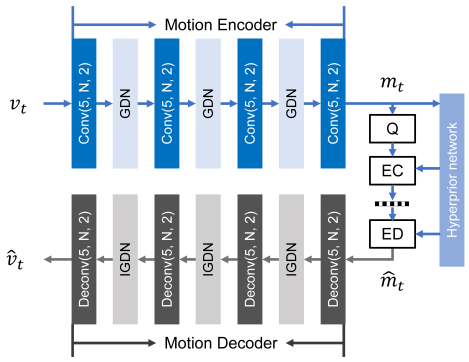

Fig. 3. Illustration of the motion encoder and decoder.

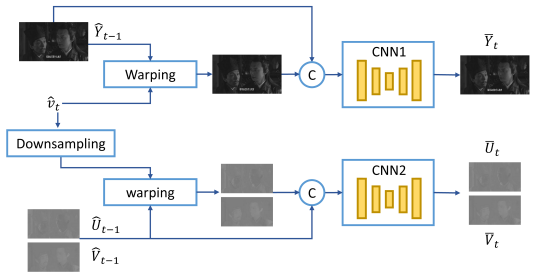

Fig. 4. Illustration of the motion compensation module.

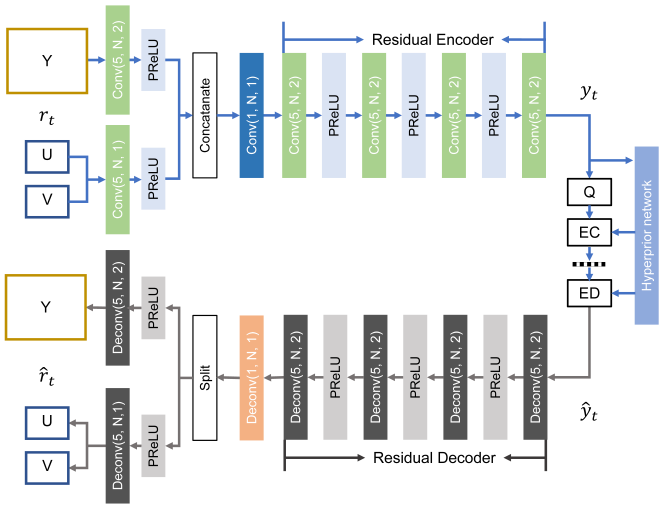

Fig. 5. Illustration of the residual encoder and decoder.

associated reference frame. As depicted in Fig. 4, the warping frame and the reconstructed optical flow frame $\hat{v}_t$ are fed to the enhancement network to compensate the luma channel. Regarding the chroma components, the downsampled optical flow is utilized to yield the warping frame. The warping result concatenated with the downsampled optical flow is subject to the enhancement network to enhance prediction.

D. Residual Encoder-Decoder on YUV420

The residuals of Y, U, and V components are calculated separately. To support the compression of YUV420 color format, the residual compression method utilizes two head-branches to deal with the luma and the chroma components. As shown in Fig. 5, to align the resolution of the features extracted from different branches, the dimension of the luma map will be reduced and the resolution of the chroma map remains unchanged. For exploiting the relationship between Y and UV components, the 1×1 convolutional layer is utilized and the features combined the cross-component information are passed to the main body of the residual encoder. At the last stage of the residual decoder, the reconstructed features are split into two branches to produce the reconstructed Y, U and V components.

E. Bit rate Estimation

Bit rate estimation is essential in the end-to-end compression framework. The mean-scale entropy model in [24] is employed in the proposed method, which assumes the distribution of the quantized latent code in each channel and position follows the Gaussian distribution with its specific mean and scale. Within such assumption, the entropy of the quantized latent code could be calculated, such that the coding bit rate could be estimated.

III. Experimental Results

A. Experiments Setup

The proposed method is developed upon the CompressAI [25] project, which is a Pytorch library for learning based compression. The Vimeo-90k [26] is served as the training dataset wherein videos in RGB color format are converted to the YUV420 format by FFmpeg [27]. The proposed end-to-end compression networks are systematically trained with the YUV420 video data. More specifically, we separately pre-train the motion estimation module, motion encoder-decoder, as well as the residual encoder-decoder. Then, all the modules are combined and jointly trained from the pre-trained parameters, aiming to minimize the loss function considering the distortion and the rate, which can be described as follows,

$$L = \lambda D + R, \tag{1}$$

where D denotes the coding distortions which are measured by the weighted mean square error (MSE) of the Y, U and V components as follows,

$$D = (6MSE_Y + MSE_U + MSE_V)/8. \tag{2}$$

R denotes the bit rate which can be calculated as the combination of the motion bit rate R_{mv} and the residual bit rate R_{res},

$$R = R_{mv} + R_{res}. \tag{3}$$

Herein, λ is used to balance the penalty of the coding distortion and the bit rate consumption, where multiple rate points can be achieved by adjusting the λ. In particular, four models are prepared with different λ values $\{0.05, 0.025, 0.00625$ and $0.0025\}$, aligning to the bit rate realized by the released models

TABLE I
PERFORMANCE COMPARISONS IN TERMS OF BD-RATE (ANCHOR: DVC).

Sequence	DVC					Proposed					YBDBR	CBDBR
	Bitrate	YPSNR	UPSNR	VPSNR	CPSNR	Bitrate	YPSNR	UPSNR	VPSNR	CPSNR		
Validation1	6257.65	42.52	46.59	50.40	44.02	2973.78	43.48	42.55	46.3	43.71	-62.14%	**-33.17%**
	3271.28	41.59	45.83	49.49	43.1	1565.67	42.28	42.21	44.31	42.53		
	1748.99	40.58	44.99	48.45	42.11	691.73	40.47	40.41	44.05	40.91		
	1045.59	39.05	44.06	47.07	40.68	380.66	38.39	40.96	43.92	39.40		
Validation2	5963.40	42.06	50.83	52.05	44.41	5231.51	43.28	44.26	46.46	43.80	-17.71%	27.62%
	3634.89	40.85	49.66	50.89	43.20	3460.61	41.59	43.33	45.34	42.28		
	2351.91	39.56	48.14	49.63	41.89	1767.89	38.99	41.80	44.39	40.01		
	1489.09	37.86	46.68	48.07	40.24	1067.71	36.71	41.31	44.05	38.20		
Validation3	8356.04	41.76	48.16	50.58	43.66	7137.31	43.02	42.93	45.27	43.29	-13.10%	21.36%
	5192.01	40.50	47.16	49.56	42.47	4972.11	41.40	42.11	43.49	41.75		
	3524.84	39.27	45.77	48.35	41.21	2665.48	38.12	39.61	42.45	38.84		
	2299.93	37.53	44.68	47.00	39.60	1697.45	35.57	38.57	40.89	36.61		
Validation4	10258.34	42.72	47.56	45.76	43.71	7024.19	45.22	40.94	40.35	44.08	-37.89%	**-17.32%**
	6414.99	41.94	46.27	44.37	42.79	5050.21	43.58	40.01	38.58	42.51		
	4808.69	40.43	44.25	42.35	41.14	2873.73	40.13	37.74	36.17	39.33		
	2974.44	39.11	42.56	40.67	39.74	1958.39	37.26	33.84	32.78	36.27		
Validation5	12068.61	35.86	44.43	42.43	37.75	11907.19	39.52	39.63	38.83	39.45	-71.27%	**-43.23%**
	6450.48	34.76	44.00	41.79	36.79	7718.11	37.11	38.81	38.19	37.46		
	4270.31	33.82	42.67	40.78	35.80	2528.43	33.43	38.48	37.69	34.59		
	2607.11	32.85	41.99	40.11	34.90	5384.39	30.52	36.43	35.61	31.90		
Validation6	3681.78	37.65	51.63	51.58	41.14	2452.39	45.77	45.17	48.53	46.04	——	**-84.23%**
	2296.61	37.37	50.40	49.56	40.53	1682.48	43.21	44.83	45.89	43.75		
	1436.93	37.15	50.07	50.19	40.39	772.81	41.24	43.16	46.10	42.09		
	931.24	36.63	49.13	48.58	39.69	439.43	38.95	43.75	46.54	40.50		
Validation7	10316.64	36.04	43.69	47.20	38.39	9581.96	40.85	42.35	42.71	41.27	-49.27%	**-31.49%**
	6385.37	35.29	43.16	46.17	37.64	6227.64	38.63	41.53	41.77	39.38		
	4144.63	34.42	42.48	44.70	36.71	2969.19	35.45	39.73	40.26	36.59		
	2600.34	32.53	41.07	42.65	34.86	1724.63	33.15	38.82	39.53	34.66		
Validation8	5273.74	37.12	48.35	50.02	40.14	3107.20	44.50	43.65	46.01	44.58	——	**-69.02%**
	3127.13	36.74	47.64	49.24	39.66	2117.35	42.25	42.9	44.19	42.57		
	1976.46	36.31	46.70	48.37	39.12	1083.18	39.72	41.39	43.90	40.45		
	1308.83	35.38	45.39	46.74	38.05	653.76	37.39	41.44	43.46	38.66		
Validation9	6742.93	42.07	50.73	49.35	44.07	5880.32	43.35	44.43	44.49	43.62	-4.61%	30.30%
	4448.27	40.99	49.70	48.24	42.99	4046.11	41.06	43.26	42.51	41.51		
	2995.89	39.52	48.33	46.49	41.49	2189.70	37.89	40.83	40.65	38.60		
	1988.04	37.93	47.05	44.72	39.92	1314.80	35.20	41.54	40.13	36.61		
Validation10	3892.08	45.13	52.29	52.73	46.97	2415.66	46.69	45.11	46.95	46.53	-42.39%	0.98%
	2430.19	44.39	51.54	51.64	46.19	1485.89	44.89	44.81	45.27	44.92		
	1493.41	43.38	50.60	50.18	45.13	698.50	42.38	42.70	44.76	42.72		
	989.93	42.11	49.08	48.67	43.80	408.01	39.81	43.79	44.86	40.94		
Avg.												**-19.82**%

of OpenDVC [28]. It is worth mentioning that the OpenDVC exhibits similar performance to the reported results in [17]. The distortion loss D represents the weight sum of the MSE of Y, U and V components.

B. Performance Comparisons with DVC

The testing sequences from Validation1 to Validation10 are provided by the Grand Challenge on Neural Network-based Video Coding in ISCAS 2022. The compression performances are measured by utilizing the component-wise BD-Rate [29] on the combined PSNR,

$$CPSNR = (6YPSNR + UPSNR + VPSNR)/8. \quad (4)$$

The group of pictures (GoP) is set to 10, which is identical to the DVC configuration. Moreover, the I frame in each GOP is encoded by VVC Test Model version 14.0 (VTM-14.0) [30] under the AI configuration. The proposed method is responsible for the P-frame compression, where the first 32 frames of each sequences are involved for test. As shown in Table I, the proposed method achieves 19.82% BD-Rate reductions on average in terms of CPSNR when compared with the DVC method. The proposed method outperforms DVC in the Y component for all sequences, especially on Validation6 and Validation8, where the associated BD-Rate could not be measured since the YPSNR at the lowest bit rate of the proposed method is larger than the YPSNR at the highest bit rate of DVC, indicating the superior performance of the proposed method on the luma component. It is worth noting that the proposed method may be inferior to DVC on some sequences regarding the chroma components.

IV. CONCLUSIONS

In this paper, a deep video compression framework for P-frame in sub-sampled color spaces has been designed. Successive modules which adapt to the YUV420 format have been systemically built and jointly optimized under the constraint of a proper weighted loss function, leading to the improvment of the compression performance. The motion estimation in sub-sampled color spaces is performed to obtain the motion information between frames, which is further compressed by a delicate motion encoder-decoder to release the transmission overhead. The decoded motion information and the previous decoded frame are leveraged by motion compensation module to produce pleasing prediction. The residual encoder-decoder utilizes and fuses cross-components information to realize the superior residual compression in sub-sampled color spaces. Experimental results show that the proposed method has improved the coding performance when compared to the DVC.

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, 2012.

[2] 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)*, 2019, pp. 1–5.

[3] 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*, vol. 109, no. 9, pp. 1463–1493, 2021.

[4] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1683–1698, 2019.

[5] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, 2018.

[6] Y. Li, L. Li, Z. Li, J. Yang, N. Xu, D. Liu, and H. Li, "A hybrid neural network for chroma intra prediction," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 1797–1801.

[7] Y. Hu, W. Yang, M. Li, and J. Liu, "Progressive spatial recurrent neural network for intra prediction," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3024–3037, 2019.

[8] Z. Zhao, S. Wang, S. Wang, X. Zhang, S. Ma, and J. Yang, "Enhanced bi-prediction with convolutional neural network for High-Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3291–3301, 2019.

[9] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced motion-compensated video coding with deep virtual reference frame generation," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4832–4844, 2019.

[10] C. Ma, D. Liu, X. Peng, Z.-J. Zha, and F. Wu, "Neural network-based arithmetic coding for inter prediction information in HEVC," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[11] K. Lin, C. Jia, Z. Zhao, L. Wang, S. Wang, S. Ma, and W. Gao, "Residual in residual based convolutional neural network in-loop filter for AVS3," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.

[12] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma, "Content-aware convolutional neural network for in-loop filtering in High Efficiency Video Coding," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3343–3356, 2019.

[13] Y. Li, L. Zhang, and K. Zhang, "Convolutional neural network based in-loop filter for VVC intra coding," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 2104–2108.

[14] R. Lin, Y. Zhang, H. Wang, X. Wang, and Q. Dai, "Deep convolutional neural network for decompressed video enhancement," in *2016 Data Compression Conference (DCC)*, 2016, pp. 617–617.

[15] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[16] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7939–7948.

[17] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 006–11 015.

[18] O. Rippel and L. Bourdev, "Real-time adaptive image compression," *arXiv preprint arXiv:1705.05823*, 2017.

[19] A. Habibian, T. V. Rozendaal, J. Tomczak, and T. Cohen, "Video compression with rate-distortion autoencoders," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7032–7041.

[20] J. Lin, D. Liu, H. Li, and F. Wu, "M-LVC: Multiple frames prediction for learned video compression," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3543–3551.

[21] G. Lu, C. Cai, X. Zhang, L. Chen, W. Ouyang, D. Xu, and Z. Gao, "Content adaptive and error propagation aware deep video compression," 2020.

[22] H. E. Egilmez, A. K. Singh, M. Coban, M. Karczewicz, Y. Zhu, Y. Yang, A. Said, and T. S. Cohen, "Transform network architectures for deep learning based end-to-end image/video coding in subsampled color spaces," *arXiv preprint arXiv:2103.01760*, 2021.

[23] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2720–2729.

[24] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *arXiv preprint arXiv:1809.02736*, 2018.

[25] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "Compressai: a pytorch library and evaluation platform for end-to-end compression research," *arXiv preprint arXiv:2011.03029*, 2020.

[26] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[27] "FFMPEG homepage," http://ffmpeg.org/.

[28] R. Yang, L. Van Gool, and R. Timofte, "OpenDVC: An open source implementation of the DVC video compression method," *arXiv preprint arXiv:2006.15862*, 2020.

[29] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves (VCEG-M33)," in *VCEG Meeting (ITU-T SG16 Q. 6)*, 2001, pp. 2–4.

[30] "VVC and VTM homepage," https://jvet.hhi.fraunhofer.de/.