IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 December 2023, accepted 2 January 2024, date of publication 8 January 2024, date of current version 6 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350643

RESEARCH ARTICLE

HDVC: Deep Video Compression With Hyperprior-Based Entropy Coding

YUSONG HU[1], CHEOLKON JUNG[1], (Member, IEEE), QIPU QIN[1], JIANG HAN[1], YANG LIU[2], AND MING LI[2]

[1]School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2]Guangdong OPPO Mobile Telecommunications Corporation Ltd., Dongguan 523860, China

Corresponding author: Cheolkon Jung (zhengzk@xidian.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62111540272.

ABSTRACT In this paper, we propose deep video compression with hyperprior-based entropy coding, named HDVC. The proposed method is based on the deep video compression (DVC) framework that replaces traditional block-based video compression with end-to-end video compression based on deep learning, aiming to improve compression efficiency and reduce computational complexity while maintaining visual quality. Based on the DVC framework, we introduce hyperprior-based entropy coding into motion compression and optimize motion vector estimation (i.e. optical flow estimation) using window attention and fast residual channel attention. Moreover, we introduce residual channel attention intermediate module into both encoding and decoding to enhance residuals and the quality of reconstructed frames. We adopt hyperprior-based entropy coding in residual compression to model feature distribution. Besides, we use learned image compression for intraframe coding based on fast residual channel attention network to generate reference frames. Experimental results show that the proposed method achieves better PSNR and MS-SSIM performance than both traditional block-based and recent deep learning-based video compression methods on UVG dataset.

INDEX TERMS Hyperprior, entropy coding, learned video compression, deep learning, end-to-end.

I. INTRODUCTION

Video compression plays a crucial role in today's digital media era. Since video contents account for the vast majority of internet traffic, efficient video compression is necessary to generate higher quality frames under a given bandwidth, significantly improving video transmission speed and viewing experience. Moreover, video compression can also be applied to action recognition [1] and model compression [2], while making video transmission and processing more efficient, saving bandwidth and storage space. Therefore, video compression is of irreplaceable importance in the field of digital media.

In the past few decades, traditional video coding standards such as HEVC [3] and VVC [4] have used prediction, transformation, quantization, and entropy coding based on block partitioning to solve complex video encoding problems. Although these standards have achieved excellent compression efficiency, they also have the following problems: 1) Each submodule relies on manual design, making it difficult to optimize the codec from a holistic perspective. 2) With the emergence of new digital media such as 360-degree panoramic videos and virtual reality (VR) videos, traditional video compression techniques are unable to meet the demands for high resolution, high frame rate, and low latency.

The advent of deep learning has created a new wave in image and video compression by end-to-end learning [5], [6], [7], [8], [9], [10]. Compared with traditional block-based video compression, deep learning-based image and video compression can achieve higher data compression rates while maintaining visual quality. Ballé et al. [11] made a groundbreaking connection between image compression and hyperprior-based model, which led to the end-to-end image compression. Inspired by the success of deep learning-based image compression, researchers have begun to explore

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Callico.

using deep neural networks to reduce temporal and spatial redundancy in videos. Up to the present, a number of deep learning-based video encoders and decoders have been proposed, which can be categorized into two main groups: P-frame compression strategies with unidirectional reference and B-frame compression strategies with bidirectional reference. For P-frame compression, researchers have proposed various methods, including PMCNN [12] based on motion extension and hybrid prediction networks, NVC [13] based on joint spatio-temporal priors aggregation, and DVC [14] based on end-to-end deep video compression to replace traditional video coding framework. These methods used motion vector estimation such as optical flow to represent temporal information of the video and compress optical flow and residuals using variational autoencoder (VAE). For B-frame compression, researchers have also proposed a series of strategies. Yang et al. [15] designed allocation strategies and recursive enhancement, Wu et al. [16] designed interpolation-based video compression networks, and Djelough et al. [17] implemented an optical flow compression networks that simultaneously decoded optical flow and interpolation coefficients.

In this paper, we propose deep video compression with hyperprior-based entropy coding, named HDVC. HDVC adopts hyper-based entropy coding in motion compression and residual compression to improve video coding efficiency. HDVC employs the DVC framework proposed by Lu et al. [14] as the backbone network, which utilizes hyperprior-based model, window attention mechanism, and fast residual channel attention network (FRCAN) module to enhance optical flow estimation for motion compression. Furthermore, HDVC uses end-to-end image compression to compress intraframes. We design a residual channel attention intermediate module and applies it to the residual compression to recover the high-frequency residual information lost by compression. Experimental results demonstrate that HDVC achieves better PSNR and MS-SSIM values on the UVG dataset than traditional video compression methods (H.264 [18] and H.265 [3]) and most deep learning-based video compression methods. Fig. 1 illustrates the network structure of HDVC.

Compared with existing methods, main contributions of this paper are as follows:

- We propose an end-to-end video compression network based on DVC framework [14], called HDVC. HDVC adopts hyperprior-based entropy coding for motion compression to generate accurate optical flow vectors. Moreover, HDVC uses a hyperprior-based entropy module for residual compression based on residual channel attention intermediate module and window attention mechanism. HDVC remarkably enhances the learning capability by considering data distribution, thus resulting in outstanding reconstruction performance.
- We introduce FRCAN module and window attention mechanism into motion compression to enhance the optical flow estimation, thus resulting in better motion compensated prediction.
- We propose a residual channel attention intermediate module in both motion compression and residual compression to improve the accuracy of the predicted and reconstructed frames. To the best of our knowledge, this is the first work of jointly applying the combination of residual network and channel attention to the learned video compression task.

The rest of this paper is as follows. In Section II, we review recent works related to end-to-end video compression and optical flow estimation. In Section III, we specifically introduced the detailed structure of each module in HDVC. In Section IV, we presented the objective experimental results and subjective visual evaluation images of HDVC. We draw conclusions of this paper in Section V.

II. RELATED WORK

A. LEARNED VIDEO COMPRESSION

With the development of deep learning-based image compression techniques, more and more end-to-end video compression methods have emerged. In this section, we will select and introduce some milestone and important papers.

Lu et al. proposed a deep learning-based video compression framework, called Deep Video Compression (DVC) [14]. DVC is derived from the HEVC [3], and followes the predictive-residual coding form used in the HEVC framework. It replaces the block-based motion estimation and motion compensation modules used in traditional video compression methods with optical flow estimation networks and optical flow alignment networks, and uses the aligned optical flow as the prediction for the current frame. DVC also utilizes image compression encoding and decoding networks [5] to compress residual information, and employs differentiable quantization and entropy coding methods [11] that meet the learning criteria. Additionally, to reduce bitrate, DVC applies encoding and decoding networks for optical flow maps that are the same as those used in image compression. By replacing every module in the predictive-residual coding framework with deep learning networks, DVC has achieved the first end-to-end video compression coding algorithm based on deep learning.

In DVC, the method of optical flow motion estimation provides a motion vector for each pixel, which enhances the accuracy of the predicted frame during motion compensation. However, due to the similarity in motion between neighboring pixels, the motion vector map contains a significant amount of redundant information. This redundancy reduces the rate-distortion performance during compression encoding. To improve the performance of the DVC architecture, Hu et al. [19] proposed a method called Resolution-adaptive Flow Coding (RaFC) to efficiently compress the motion information of video compression. This method consists of two new approaches: RaFC frame at the frame level and RaFC block at the block level. RaFC frame can process complex or simple motion patterns globally by automatically selecting

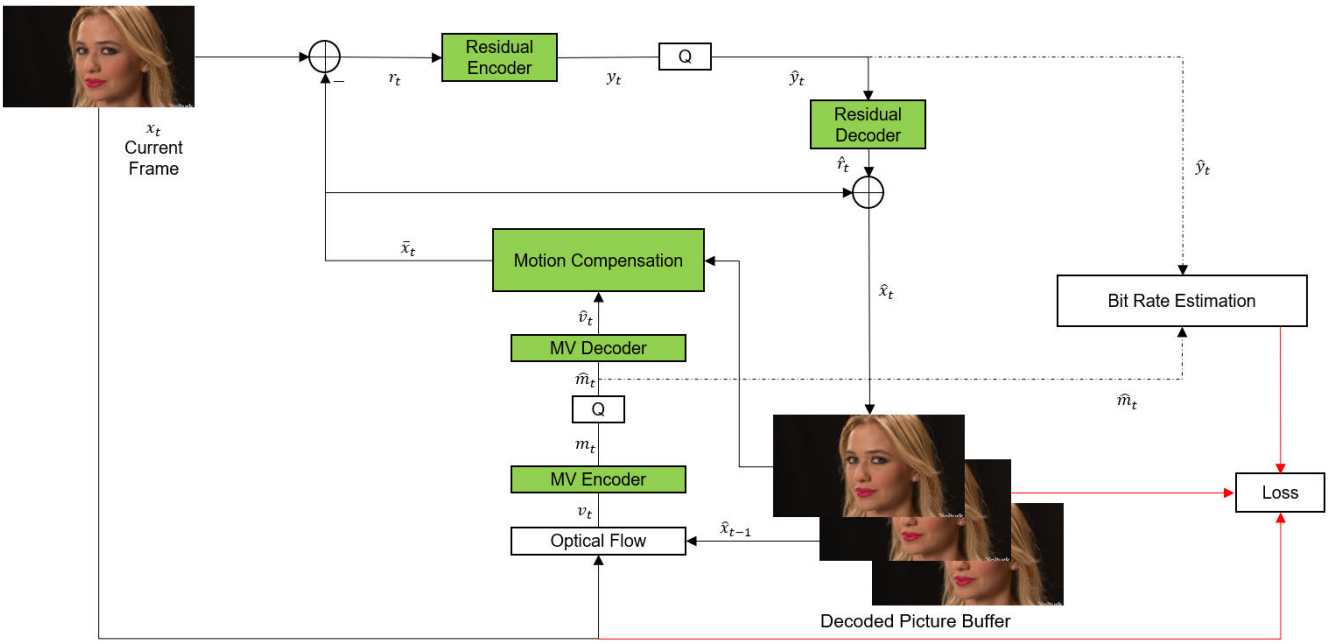

FIGURE 1. Network structure of the proposed Hyperprior based Deep Video Compression (HDVC). The green parts represent the improvements we make to the DVC framework [14]. x_t is the current frame, $\hat{x}_t$ is the reconstructed frame, $\hat{x}_{t-1}$ is the previous reconstructed frame, and $\bar{x}_t$ is the predicted frame. v_t is the motion information and $\hat{v}_t$ is the reconstructed motion information. m_t and $\hat{m}_t$ are the quantized motion representation before and after compression, respectively. y_t and $\hat{y}_t$ refer to the quantized residual representations prior to and following compression, correspondingly. r_t is the residual to be encoded, and $\hat{r}_t$ is the reconstructed residual.

the best resolution from a multi-scale flow map, while RaFC block can process different types of motion patterns locally by choosing the optimal resolution for multi-scale motion features at each block. Moreover, the authors used an optical flow compression network with adaptive scaling and a neural network trained to adaptively select the scaling size of the optical flow map during the compression of the motion vector map. This reduced redundancy in the optical flow map and improved the rate-distortion performance.

Due to the artifacts caused by motion compensation based on optical flow in the predicted frames, the prediction error is increased, which reduces the rate-distortion performance of subsequent residual compression in DVC. To improve the accuracy of optical flow and reduce artifacts, Lu et al. [20] proposed a content-adaptive and error-propagation-aware video compression system. By considering the compression performance of multiple consecutive frames instead of a single frame, their method effectively alleviates error propagation in reconstructed frames. More importantly, the proposed system designs an online encoder update scheme instead of using manually crafted encoding modes in traditional compression systems. As a result, the encoder can adapt to different video contents and achieve better compression performance by reducing the domain gap between the training and testing datasets.

Lin et al. proposed an end-to-end learning-based video compression scheme, named Multiple Frames Prediction for Learned Video Compression (M-LVC) [21], which is suitable for low-latency scenarios. Specifically, M-LVC leverages the correlation between multiple frames by jointly optimizing multi-frame prediction, motion vector prediction, and residual compression to improve video compression performance. In addition, M-LVC introduces a new frame sequence rearrangement method to better utilize the information of multi-frame prediction, achieving higher compression efficiency and better visual quality throughout the entire compression process.

B. OPTICAL FLOW ESTIMATION

Optical flow estimation is a fundamental task in computer vision, aiming to determine the motion speed and direction of each pixel in a sequence of images. It has wide applications, such as video compression, motion analysis, object tracking, and robot navigation. Currently, optical flow estimation methods can be mainly divided into two categories: traditional methods and deep learning methods. Traditional methods are mainly based on the assumption of brightness consistency between pixels and calculate optical flow by finding the motion field that minimizes the changes of gray values between two images. In contrast, deep learning methods use neural networks to learn the optical flow relationship between input images, which has the advantage of learning richer features and achieving better accuracy and robustness in optical flow estimation. Currently, deep learning-based optical flow estimation methods have been widely used in the field of video compression.

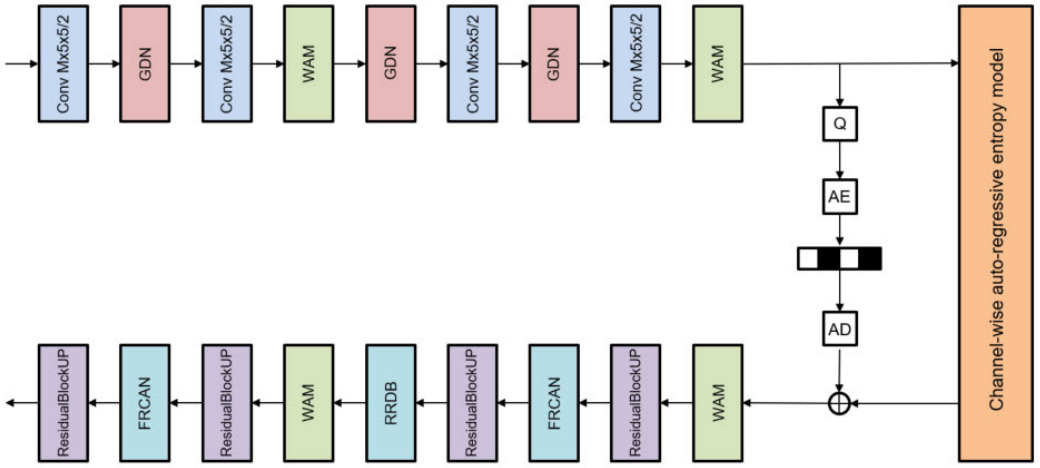

FIGURE 2. Network structure of the proposed intraframe coding. GDN: Generalized divisive normalization. WAM: Window attention mechanism. FRCAN: Fast residual channel attention network. RRDB: Residual in residual dense block. Q: Quantization. AE: Arithmetic encoder. AD: Arithmetic decoder.

Ranjan et al. [22] proposed an optical flow estimation method based on Spatial Pyramid Network (SPyNet), which can effectively utilize multi-scale information in images to improve the accuracy of optical flow estimation. Specifically, SPyNet uses multiple convolutional layers and pooling layers to extract features from input images, and then fuses these features through upsampling and convolutional operations to obtain the final optical flow estimation result. Compared with other optical flow estimation methods, SPyNet has higher accuracy and better robustness. In addition, the method has lower computational cost and can be used in real-time applications.

Recurrent all-pairs Field Transforms (RAFT) [23] is a deep learning-based optical flow estimation method proposed by Teed et al. in 2020. Compared to previous optical flow computation frameworks, Raft uses high-resolution maintenance and updating of a single fixed flow field, which achieves breakthroughs such as reducing prediction error rates and decreasing the probability of missing small and fast-moving targets. Additionally, previous optical flow computation frameworks did not limit the weights between iterations, leading to restrictions on iteration counts. In contrast, Raft's update operator is periodic and lightweight, with only 2.7 million parameters, which can iterate over 100 times. Furthermore, while the fine-tuning modules in previous frameworks usually only use regular convolution or correlation layers, Raft uses a newly designed update operator composed of convolutional Gate Recurrent Units (GRUs), which performs better on 4D multiscale correlated vectors.

III. PROPOSED METHOD

As depicted in Fig. 1, HDVC first performs optical flow estimation from the current and reference frames to estimate motion information. The motion information (optical flow vectors) is compressed into bitstream by the motion compression network (MV encoder and decoder), followed by using the motion compensation network to produce a predicted frame based on the decoded optical flow information and reference frame. Then, HDVC subtracts this predicted frame from the current frame to obtain residual information. The residual information is compressed by the residual compression network (residual encoder and decoder), while generating another bitstream. Finally, the decoded residual information is combined with the predicted frame to obtain a reconstructed frame.

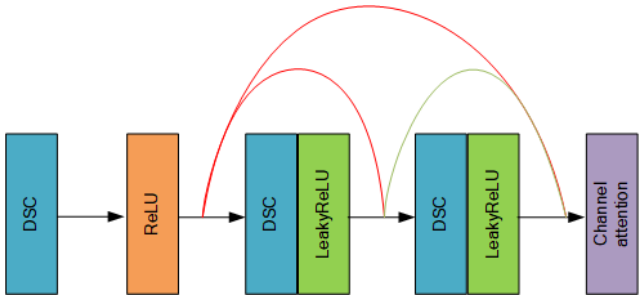

FIGURE 3. Network structure of the proposed residual channel attention block (RCAB). We combine four RCABs to form the fast residual channel attention network (FRCAN) module in Fig. 2 to recover and select important channel features. DSC: Depthwise separable convolution.

A. INTRAFRAME CODING

The intraframe coding has an asymmetric network structure based on learned image compression that consists of simple encoding and complex decoding to consider limited bandwidth. As shown in Fig. 2, simple encoding improves the encoding speed and reduces the bitstream, while complex decoding compensates for the information lost by compression and improves the quality of decoded images. We use depthwise separable convolution (DSC) to replace standard convolution in residual learning and attention module. Since DSC needs nearly one third of parameters in

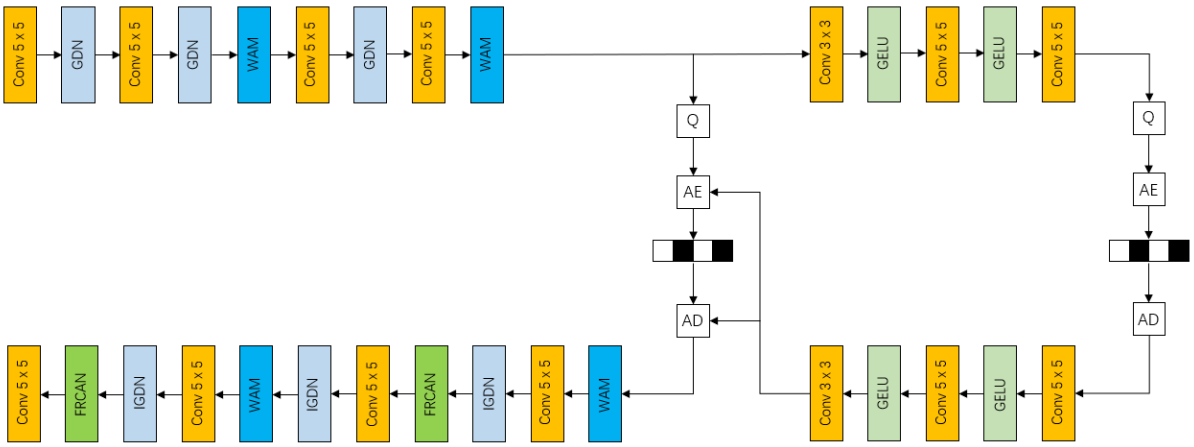

FIGURE 4. Network structure of motion compression. GDN: Generalized divisive normalization. WAM: Window attention mechanism. FRCAN: Fast residual channel attention network. Q: Quantization. AE, AD represent arithmetic encoder and arithmetic decoder, respectively.

standard convolution, DSC achieves speedup of the network with stability. As shown in Fig. 3, we design a residual channel attention block (RCAB) for image reconstruction that combines DSC with residual attention. The residual attention reduces bitstream and captures informative features, thus leading to improvement of both runtime and visual quality. We combine four RCABs to form the fast residual channel attention network (FRCAN) module in Fig. 2, and select important channel features.

B. MOTION COMPRESSION

As shown in Fig. 4, the network structure of motion compression is similar to that of end-to-end image compression. We introduce hyperprior-based entropy coding into motion compression to better describe the distribution of motion vectors (i.e. optical flow vectors) and minimize information loss while ensuring compression rate. Moreover, we replace the Rectified Linear Unit (ReLU) activation function in entropy coding with Gaussian Error Linear Unit (GELU) [24] to address the problem of gradient explosion. Inspired by the success of the previous image compression network, we enhance capabilities of the feature extraction and reconstruction in the motion compression network by combining the window attention mechanism proposed by Zou et al.'s [25] with the FRCAN module based on RCAB in Fig. 3. The combination allows the decompressed motion vectors to more accurately capture dynamic details and improve the visual quality of the video. Since HDVC adopts a holistic end-to-end learning approach and needs to train multiple networks simultaneously, we use simple convolutional neural networks (CNNs) to upsample and downsample the motion vectors while saving computational resources.

C. RESIDUAL COMPRESSION

In video compression, there is a high degree of similarity between predicted frames and adjacent frames, which means that low-frequency residual information is usually well-preserved in the predicted frames. Therefore, the residual compression network needs to do efficient compression and transmission of high-frequency residual information. To achieve this goal, we use a symmetrically compressed autoencoder structure as illustrated in Fig. 5. The network incorporates a large number of residual and attention mechanisms to filter and enhance residual information before compression during the encoding stage. These mechanisms are also utilized during decoding to recover and select residual information, resulting in higher quality of reconstructed frames. To enhance the feature extraction capability of the entropy encoder, residual modules and attention mechanisms are introduced into the entropy encoding module of the residual compression network. It improves visual quality in compression by capturing structural probability distribution of residual information effectively. Furthermore, due to the limited computational resources, it is not feasible to use a large-scale dense residual network simultaneously at the encoding and decoding stages to enhance and recover residual information. To tackle this problem, we propose a residual channel attention intermediate module (RCAIM) to improve the residual information. As shown in Fig. 6, we integrate channel attention into a residual network. The pipeline of this module is as follows: First, the input information is convolved by the convolution layer to generate more features. Then, the channel attention mechanism assigns weights to these features and combines them with the input information. The process selectively preserves or eliminates residual information from input, which enhances and restores crucial residual features.

In the motion compression network, the optical flow information contains both low-frequency and high-frequency features. Thus, it is necessary to consider both frequency features during motion compression for recovery. To address this issue, we build an asymmetric framework for motion

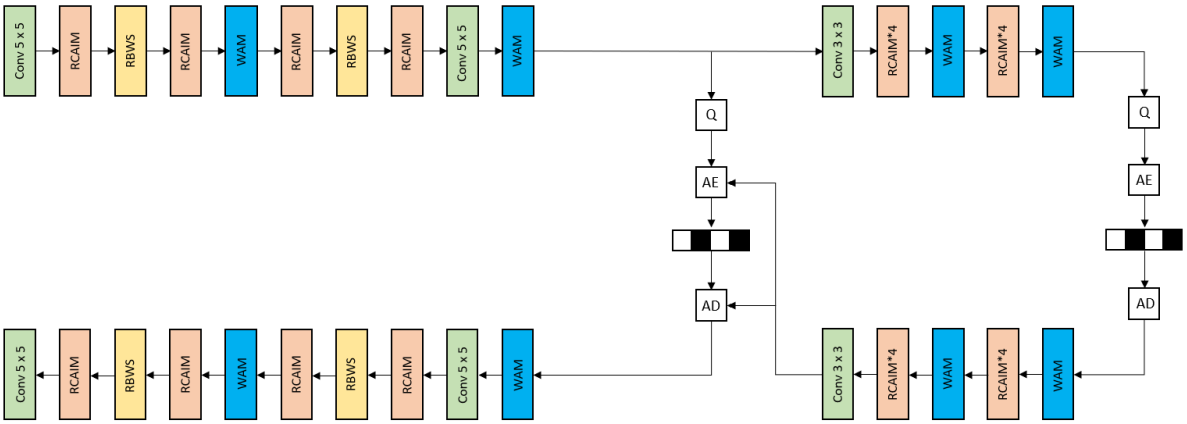

FIGURE 5. Network structure of residual compression. WAM: Window attention mechanism. Q: Quantization. AE, AD represent arithmetic encoder and arithmetic decoder, respectively. RCAIM: Residual channel attention intermediate module. RBWS: Residual block with stride.

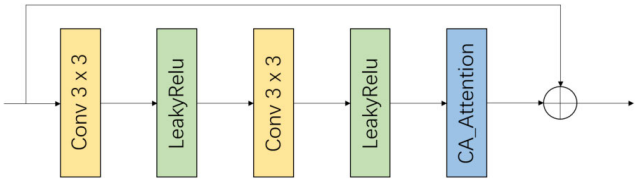

FIGURE 6. Network structure of residual channel attention intermediate module (RCAIM).

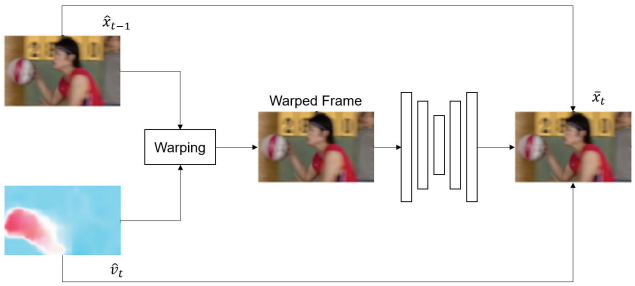

FIGURE 7. Illustration of motion compensation. We replace the residual layers in CNN with the RCAIM module.

compression that takes the complexity of the encoding process into account. However, the residual information is primarily high-frequency features. Therefore, we build a symmetric network for residual compression that incorporates residual attention to pay more attention to the high-frequency information at the input while restoring the lost high-frequency features at the output, thus leading to better decoded residual information.

D. MOTION COMPENSATION

We adopt the motion compensation in DVC framework [14] by replacing the residual layers in CNN with the RCAIM module. As shown in Fig. 7, the motion compensation utilizes motion vectors to warp the reference frame. Subsequently, the warped frame, reference frame, and motion vectors are put to a CNN module to generate the predicted frame. To enhance the accuracy of the predicted frame, we replace the ordinary residual layers in CNN with the proposed RCAIM module to generate and retain crucial features for frame prediction.

E. LOSS FUNCTION

HDVC reduces the number of bits in encoding while minimizing the degree of distortion between the original input frame x_t and the reconstructed frame $\hat{x}_t$. To achieve this, we formulate the rate-distortion optimization problem as follows:

$$\lambda D + R = \lambda d(x_t, \hat{x}_t) + (H(\hat{m}_t) + H(\hat{y}_t)) \tag{1}$$

We utilize $d(x_t, \hat{x}_t)$ to measure the degree of distortion between the original input frame x_t and the reconstructed frame $\hat{x}_t$ and adopt mean squared error (MSE) to calculate it. The size of the bitstream required during encoding is represented by $H(\cdot)$. In accordance with DVC [14], we use the number of bits required to encode the residual representation $\hat{y}_t$ and the motion representation $\hat{m}_t$ as the encoding bitstream. Furthermore, we use the Lagrange multiplier λ to balance the bit rate and output frame quality. The loss function is obtained by the reconstructed frame $\hat{x}_t$, the original frame x_t, and the estimated bitstream as illustrated in Fig. 1.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

1) TRAINING

We train HDVC with different λ values (λ = 256, 512, 1024, 2048) using the Vimeo-90k dataset [26]. HDVC is implemented in Pytorch framework, which takes about 7 days to train the whole network using RTX 3090 GPU. All models are trained for 1.1M iterations using the Adam optimizer [27] with a batch size of 8. Although the RTX 3090 GPU memory (24GB) can accommodate the maximum batch size of 12 for training the proposed network, we set batch size to 8 for efficient use of memory resource. For each model, the initial

TABLE 1. Comparison of average encoding and decoding time among Ballé et al. [11], Minnen et al. [5], Cheng et al. [6] and Zou et al. [25] on Kodak dataset using one RTX 3090 GPU. Note that Cheng et al.'s results are based on a lightweight implementation (without Gaussian mixture likelihoods) in CompressAI framework [30].

Method	Enc(s)	Dec(s)	PSNR(dB)	MS-SSIM	bpp
Ballé [11]	0.0250	0.0189	34.53	0.9836	0.669
Minnen [5]	2.5202	5.3006	35.09	0.9837	0.639
Cheng [6]	3.0139	5.7965	34.95	0.9838	0.595
Zou [25]	0.0884	0.0916	35.80	0.9855	0.644
Proposed	**0.0835**	**0.0979**	**36.24**	**0.9873**	**0.681**

learning rate is set to 1×10^{-4} for 75k iterations, and drops to 3×10^{-5} for another 15k iterations, 1×10^{-5} for the last 20k iterations. The resolution of training frames is 256×256. The motion estimation module is initialized with the pretrained weights in Ranjan et al.'s work [22].

2) EVALUATION

We evaluate the performance of HDVC on both the UVG dataset [28] and the VVC Class B dataset (*BQTerrace*, *BasketballDrive*, and *Cactus*) [4]. These datasets feature a diverse range of content and resolutions, which are commonly used as benchmarks for performance evaluation of video compression methods.

B. PERFORMANCE EVALUATION AND COMPARISON

1) VISUAL COMPARISON

We provide visual quality comparison of the proposed method with AVC/H.264 [29], HEVC/H.265 [3], VVC/H.266, and DVC [14]. As shown in Figs. 8 and 9, HDVC achieves better visual quality with objective evaluation indicators and fewer bitstream than them. Moreover, HDVC effectively reduces noise and blocky artifacts, and has a significant improvement in texture reconstruction (e.g. the enlarged earring in Fig. 8 and the moving player in Fig. 9).

2) INTRAFRAME CODING

As mentioned in Section III-A, we propose a learned image compression network for intraframe coding that employs an asymmetric structure in encoding and encoding. The proposed network generates a compact bitstream by a simple encoding end, while recovering the output image by a complex decoding end. As shown in Table 1, the proposed network enables better image recovery while maintaining high compression ratios with moderate encoding and decoding time.

3) QUANTITATIVE MEASUREMENTS

We provide the RD curve comparison among different methods: 1) Traditional video compression methods: HM [3], VTM [4], x264 and x265; 2) Deep-learning based video compression methods (DVC [14], DVCp [31], LU_ECCV20 [20], FVC [8] and DCVC [9] and HDVC. In traditional video compression methods, x264 and x265 are performed in the RGB color space, while HM and VTM are performed in the YUV color space. To generate the compressed frames by x264 and x265, we follow the setting in Wu et al.'s work [16] and use FFmpeg [32] with very fast mode. The GOP sizes for the UVG dataset and VVC Class B dataset are 12 and 10, respectively.

TABLE 2. BD-rate comparison among the proposed method, other deep learning-based methods and traditional methods. DVC [14] is anchor for comparison, while PSNR and MS-SSIM are measured on UVG and VVC Class B testing dateset. x264 and x265 are performed in the RGB color space, while HM and VTM are performed in the YUV color space. A positive value indicates inferior performance to the anchor, while a negative value indicates superior performance to the anchor.

	UVG		VVC Class B	
Method	PSNR	MS-SSIM	PSNR	MS-SSIM
DVC [14]	0	0	0	0
DVCp [31]	-18.81%	-18.62%	-20.09%	-19.51%
Lu [20]	-10.80%	-22.91%	-16.12%	-12.05%
FVC [8]	-29.83%	-44.65%	-23.73%	-39.18%
DCVC [9]	-36.80%	-42.85%	-39.78%	-40.31%
x264	23.54%	21.85%	22.07%	43.58%
x265	-7.56%	-13.56%	-4.32%	29.72%
HM [3]	-55.55%	-56.83%	-48.10%	-36.37%
VTM [4]	-65.97%	-64.24%	-64.33%	-56.32%
Proposed	**-32.93%**	**-40.92%**	**-6.03%**	**-21.85%**

Fig. 10 illustrates the performance of HDVC compared to the other methods on the UVG dataset [28]. When evaluating based on PSNR, HDVC outperforms DVC, DVCp, Lu et al.'s, and the others in most bit rate ranges, and only slightly falls behind DCVC, HM, and VTM at a full bit rate. However, when MS-SSIM is used as an evaluation metric, HDVC is only slightly inferior to DCVC, FVC, HM, and VTM. In comparison with the other methods, HDVC exhibits superior performance. Fig. 11 shows that HDVC decreases PSNR values slightly on the VVC Class B dataset [4]. This decrease is attributed to the low frame rate in the VVC Class B dataset, which is only 50 or 60 fps, compared to the high frame rate of 120 fps in the UVG dataset. This is mainly caused by motion estimation errors during interframe coding and error propagation in the motion compensated prediction. However, when MS-SSIM is used as the evaluation metric, HDVC outperforms all other deep learning-based video compression methods except for DCVC and FVC, and performs better than HM and VTM at high bit rates. The results prove that HDVC achieves better structural recovery in video compression.

Table 2 shows BD-rate comparison based on DVC. HDVC achieves performance improvement of approximately 32.93% and 40.92% in terms of PSNR and MS-SSIM, respectively, over DVC [14] in the UVG dataset. In the VVC Class B dataset, HDVC also shows performance improvement over DVC [14] of approximately 6.03% and 21.85% in terms of PSNR and MS-SSIM, respectively. Meanwhile, on the UVG dataset, compared to FVC, HDVC achieves performance improvement of 3.1% in terms of PSNR, while HDVC performs slightly worse about 3.73% in terms of MS-SSIM. Overall, HDVC obtains similar performance to FVC on the UVG dataset. Additionally,

FIGURE 8. Visual quality comparison of the reconstructed frame from UVG dataset.

FIGURE 9. Visual quality comparison of the reconstructed frame from VVC Class B dataset.

compared to Lu et al.'s work [20], our results demonstrate performance improvement of 12.3% in terms of MS-SSIM on the VVC Class B dataset. The experimental results show that HDVC present outstanding performance in terms of both PSNR and MS-SSIM metrics.

However, the proposed HDVC performs worse than HM and VTM in compression efficiency. This is because HDVC is based on the DVC framework [14] and newly introduces hyperprior-based entropy coding into motion compression and residual compression to improve the compression efficiency. Thus, it is obvious that HDVC achieves remarkable performance improvement over DVC [14] in terms of both PSNR and MS-SSIM. Compared with other deep learning-based methods, HDVC achieves comparable or better performance only except for DCVC [9].

C. ABLATION STUDY

We perform two ablation experiments on FRCAN module, window attention mechanism and RCAIM in HDVC. Thus, we train two groups of models: 1) We remove the window attention mechanism and FRCAN module from the proposed optical flow compression network, named as 'W/O win and FRCAN'. 2) We use the optical flow compression network and the front-end and back-end processing parts of residual compression network proposed by DVC [14], retain the residual channel attention intermediate module (RCAIM) and

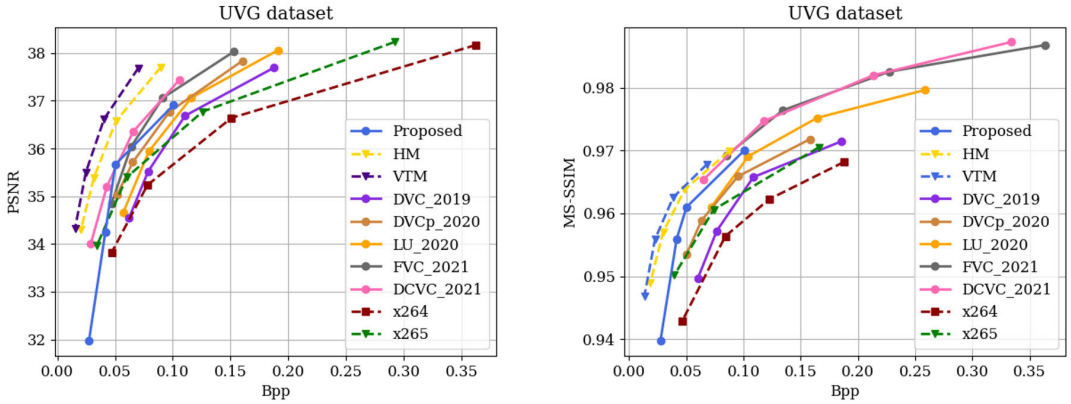

FIGURE 10. RD curves on UVG dataset. x264 and x265 are performed in the RGB color space, while HM and VTM are performed in the YUV color space.

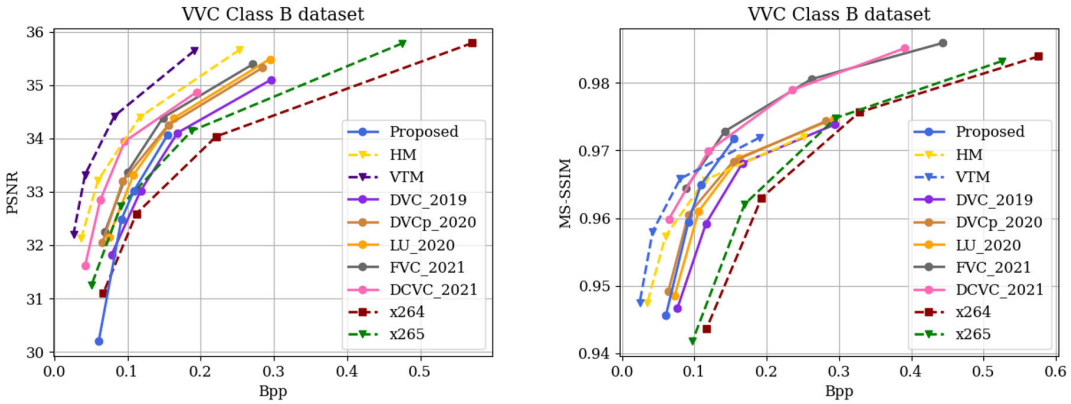

FIGURE 11. RD curves on VVC Class B dataset. x264 and x265 are performed in the RGB color space, while HM and VTM are performed in the YUV color space.

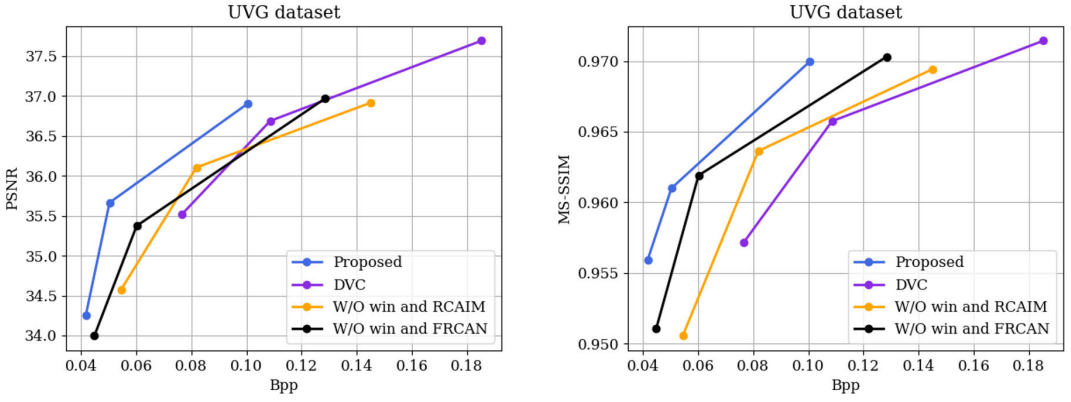

FIGURE 12. Ablation study on FRCAN module, window attention mechanism and RCAIM in HDVC. We obtain RD curves on UVG dataset.

TABLE 3. Average BD-rates on three λ values (λ = 512, 1024, 2048) on UVG dataset for ablation study.

	DVC	W/O win and FRCAN	W/O win and RCAIM	**Proposed**
PSNR	0	-7.66%	-2.34%	**-30.05%**
MS-SSIM	0	-29.53%	-12.71%	**-39.71%**

window attention mechanism in the residual compression network's entropy encoding part, named as 'W/O win and RCAIM'. Then, we compare the performance of them with HDVC to verify their effectiveness. Based on the results illustrated in Fig. 12, the network of 'W/O win and FRCAN' exhibits significant PSNR and MS-SSIM decrease at medium to high bitrates, compared with HDVC. This indicates that the integration of window attention mechanism and FRCAN module in the optical flow compression network remarkably enhances the optical flow estimation performance, resulting in more accurate predicted frames. Furthermore, the experimental results on W/O win and RCAIM show that RCAIM block and window attention mechanism in residual entropy coding can effectively restore the structural features of reconstructed frames, leading to better visual quality than DVC. Table 3 presents BD-rates on the UVG testing dataset using DVC as an anchor. 'W/O win and FRCAN' achieves bit rate reduction of 7.66% in PSNR and 29.53% in MS-SSIM, while 'W/O win and RCAIM' achieves bit rate reduction of 2.34% in PSNR and 12.71% in MS-SSIM. The experimental results indicate that two modules affects a certain degree of performance improvement, demonstrating their necessity for video compression.

V. CONCLUSION

In this paper, we have proposed HDVC, i.e. a novel deep video compression framework based on hyperprior. We have introduced hyperprior-based entropy coding and RCAIM to enhance the performance of residual compression, resulting in the improvement of reconstruction performance. Moreover, we have used hyperprior-based entropy coding and FRCAN module with window attention mechanism to improve the accuracy of motion compression, thus leading to improving the predicted frames. Furthermore, we have employed an end-to-end image compression network for intraframe coding, thus generating precise reference frames. The experimental results demonstrate that HDVC has outstanding structural recovery in video compression and outperforms the original DVC framework by 32.93% and 40.92% in both PSNR and MS-SSIM metrics, respectively.

However, HDVC shows relatively lower performance for video sequences with a low frame rate such as VVC Class B dataset. This is mainly from the error propagation in the motion compensated prediction. Therefore, our future work focuses on motion compression and motion compensated prediction to improve the compression efficiency for video sequences with a low frame rate.

REFERENCES

[1] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Compressed video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6026–6035.

[2] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[5] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[6] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[7] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, "End-to-end optimized versatile image compression with wavelet-like transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1247–1263, Mar. 2022.

[8] Z. Hu, G. Lu, and D. Xu, "FVC: A new framework towards deep video compression in feature space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1502–1511.

[9] J. Li, B. Li, and Y. Lu, "Deep contextual video compression," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 18114–18125.

[10] O. Rippel, A. G. Anderson, K. Tatwawadi, S. Nair, C. Lytle, and L. Bourdev, "ELF-VC: Efficient learned flexible-rate video coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14459–14468.

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

[12] Z. Chen, T. He, X. Jin, and F. Wu, "Learning for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 566–576, Feb. 2020.

[13] H. Liu, M. Lu, Z. Ma, F. Wang, Z. Xie, X. Cao, and Y. Wang, "Neural video coding using multiscale motion compensation and spatiotemporal context model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3182–3196, Aug. 2021.

[14] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10998–11007.

[15] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6627–6636.

[16] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 416–431.

[17] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6420–6428.

[18] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[19] Z. Hu, Z. Chen, D. Xu, G. Lu, W. Ouyang, and S. Gu, "Improving deep video compression by resolution-adaptive flow coding," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 193–209.

[20] G. Lu, C. Cai, X. Zhang, L. Chen, W. Ouyang, D. Xu, and Z. Gao, "Content adaptive and error propagation aware deep video compression," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 456–472.

[21] J. Lin, D. Liu, H. Li, and F. Wu, "M-LVC: Multiple frames prediction for learned video compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3543–3551.

[22] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2720–2729.

[23] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 402–419.

[24] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[25] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17471–17480.

[26] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[28] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proc. 11th ACM Multimedia Syst. Conf.*, May 2020, pp. 297–302.
[29] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Mar. 2003.
[30] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "CompressAI: A PyTorch library and evaluation platform for end-to-end compression research," 2020, *arXiv:2011.03029*.
[31] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3292–3308, Oct. 2021.
[32] S. Tomar, "Converting video formats with FFmpeg," *Linux J.*, vol. 2006, no. 146, p. 10, 2006.

YUSONG HU received the B.S. degree in electronic information science and technology from Xidian University, China, in 2020, where he is currently pursuing the M.S. degree in electronic engineering. His research interests include video coding and deep learning.

CHEOLKON JUNG (Member, IEEE) is a Born Again Christian. He received the B.S., M.S., and Ph.D. degrees in electronic engineering from Sungkyunkwan University, Republic of Korea, in 1995, 1997, and 2002, respectively. From 2002 to 2007, he was a Research Staff Member with the Samsung Advanced Institute of Technology, Samsung Electronics, Republic of Korea. From 2007 to 2009, he was also a Research Professor with the School of Information and Communication Engineering, Sungkyunkwan University. Since 2009, he has been with the School of Electronic Engineering, Xidian University, China, where he is currently a Full Professor and the Director with the Xidian Media Laboratory. His main research interests include image and video processing, computer vision, pattern recognition, machine learning, computational photography, video coding, virtual reality, information fusion, multimedia content analysis and management, and 3DTV.

QIPU QIN received the B.S. degree in communication engineering from Ningxia University, China, in 2017. He is currently pursuing the Ph.D. degree in electronic engineering with Xidian University, China. His main research interests include image and video processing, video coding, and virtual reality.

JIANG HAN received the B.S. degree in communication engineering from Henan Polytechnic University, China, in 2022. He is currently pursuing the M.S. degree in electronic engineering with Xidian University, China. His main research interests include video coding and deep learning.

YANG LIU received the B.S. degree in electronic and information engineering from Hunan University, China, in 2014, and the M.S. degree in telecommunications from The Hong Kong University of Science and Technology, China, in 2016. Since 2017, she has been a Standardization Engineer with Guangdong OPPO Mobile Telecommunications Corporation Ltd., China. Her research interests include video coding and multimedia communications.

MING LI received the B.S. degree in telecommunication engineering and the Ph.D. degree in communication and information systems from Xidian University, China, in 2005 and 2010, respectively. From 2010 to 2019, he was a Senior Research Staff in standardization with ZTE Corporation, China. Since 2019, he has been a Senior Standardization Engineer with Guangdong OPPO Mobile Telecommunications Corporation Ltd., China. His research interests include video coding and multimedia communications.