Hierarchical motion-compensated deep network for video compression

Liu, Ying, Du, Pengli, Li, Yuzhu

Ying Liu, Pengli Du, Yuzhu Li, "Hierarchical motion-compensated deep network for video compression," Proc. SPIE 11730, Big Data III: Learning, Analytics, and Applications, 117300J (12 April 2021); doi: 10.1117/12.2586459

SPIE.

Event: SPIE Defense + Commercial Sensing, 2021, Online Only

Hierarchical Motion-Compensated Deep Network for Video Compression

Ying Liu, Pengli Du, and Yuzhu Li

Department of Computer Science and Engineering, Santa Clara University
Santa Clara, CA 95053, United States

ABSTRACT

Video coding is the process of reducing the huge volume of video data to a small number of bits. High coding efficiency reduces the bandwidth required for video streaming, and the space required to store the video data on electronic devices, while maintaining the fidelity of the decompressed video signal. In recent years, deep learning has been extensively applied in the field of video coding. However, it remains challenging how to explore the intra- and inter-frame correlations in deep learning-based video coding systems to improve the coding efficiency. In this work, we propose a hierarchical motion estimation and compensation network for video compression. The video frames are tagged as intra-frames and inter-frames. While intra-frames are compressed independently, the inter-frames are hierarchically predicted by adjacent frames using a bi-directional motion prediction network, which results in highly sparse and compressible residue. The residue frames are then compressed via separately trained residue coding networks. Experimental results demonstrate that the proposed hierarchical deep video compression network offers significantly higher coding efficiency and superior visual quality compared to prior arts.

Keywords: Artificial intelligence, convolutional neural network, computer vision, deep learning, motion estimation, multi-scale structural similarity, video coding, video compression.

1. INTRODUCTION

Video coding is the process that compresses video data to reduce the data amount, to save the storage memory and data transmission bandwidth. While traditional video coding methods rely on signal processing techniques such as discrete-cosine transform (DCT), quantization, and entropy coding, recent advances in deep learning have triggered the development of learning-based image compression and video coding systems.

The major difference between video coding and image compression is, video sequence is consisted of successive motion pictures that involve scene dynamics and are correlated. A video coding system can leverage the inter-frame correlations and reduce the information redundancy as much as possible. Traditional video coding standards such as H.264, HEVC, and Versatial Video Coding (VVC) adopt motion estimation and motion compensation, which has to be conducted for each individual video, thus incurs high computational complexity at the encoder side. With the availability of huge amount of training videos, deep learning has turned a video coding system into a learning paradigm, which can automatically extract features for efficient video compression at the encoder side, and recover the video frames from these highly compressed features at the decoder side.

For example, [1] replaces each component of the traditional video coding system by a convolutional neural network, such as motion estimation, motion compression, post-processing to generate the predicted inter-frame, and residue compression. Although it achieves good performance, it needs to explicitly generate the motion field and compress it, which leads to extra bit rates. Moreover, it is a frame-wise compression system that takes the full-resolution video frame as input and performs convolution to extract features, which incurs high computational complexity. In [2], the inter-frame residue is compressed by a recurrent neural network (RNN). This approach

Further author information: (Send correspondence to Ying Liu)
Ying Liu: E-mail: yliu15@scu.edu
Pengli Du: E-mail: pdu@scu.edu
Yuzhu Li: E-mail: yli11@scu.edu

Big Data III: Learning, Analytics, and Applications, edited by Fauzia Ahmad, Panos P. Markopoulos, Bing Ouyang, Proc. of SPIE Vol. 11730, 117300J

is able to make use of the correlation in a large number of frames, instead of the very limited reference frames in the non-recurrent approaches. Nevertheless, in this method, frame prediction is uni-directional. In [3], 3D convolution is adopted to exploit spatial-temporal correlations and to compress multiple frames together. In [4], video frames are divided into key frames and non-key frames. While keys frames are compressed individually by a CNN, the non-key frames are compressed by CNN-based interpolation. In [5], the temporal correlation is leveraged by optical-flow based motion estimation and residue coding. To further reduce energy, it predicts flow elements among successive frames and compresses the predictive difference of optical flows.

Other methods tried to integrate deep learning into the traditional video coding framework, which are the so-called "deep tools". For instance, in [6], [7] CNN-based predictors are integrated into HEVC to improve the accuracy of inter-frame prediction. In [8], a CNN-based fractional-pixel motion compensation is proposed and integrated into HEVC. In addition, a CNN-based residue super resolution method is proposed for video coding in [9] and integrated into HEVC.

In this work, we propose a pure CNN-based deep video coding architecture, without any elements in traditional video coding. To reduce the computational complexity and model size, we adopt a patch-based approach, in which the video frames are divided into small patches to process. Our video coding system has a hierarchical structure with a GOP size of 4. It has three layers: Layer-1 compresses the intra (I) frames, Layer-2 compresses the B_1 frames (the middle frame between two successive I frames), and Layer-3 compresses B_2 frame (the middle frame between a preceding Layer-1 I frame and a subsequent Layer-2 B_1 frame) and B_3 frame (the middle frame between a preceding Layer-2 B_1 frame and a subsequent Layer-1 I frame). In particular, the Layer-2 compression adopt a bi-directional inter-frame prediction network, followed by residue coding. A similar approach is adopted in Layer-3 coding. Instead of explicitly carrying out motion estimation and compensation, our bi-directional inter-frame prediction networks can directly generate the predicted target blocks, which eliminates the compression of motion fields. We demonstrate by experimental studies on common test video sequences that the proposed hierarchical compression model achieved a higher coding efficiency than intra-frame coding and two-layer inter-frame coding [10].

The rest of the paper is organized in the following sections: Section 2 introduces the proposed network, Section 3 demonstrates the effectiveness of our proposed method through experimental studies, and Section 4 concludes the work.

2. PROPOSED NETWORK

Fig. 1 shows the hierarchical coding structure of our proposed method. We adopt a GOP size of 4, and three layers of coding. Layer-1 compresses the I frames independently. Layer-2 compresses the B_1 frame, by first predicting it with the nearest preceding and subsequent decoded I frames. The prediction direction is given by the arrows in Fig. 1. Then the residue between the ground-truth B_1 and the prediction is compressed. Layer-3 has two parts: part-1 compresses B_2, by first predicting it from the decoded preceding I frame and the decoded subsequent B_1, followed by residue compression; while part-2 compresses B_3, by first predicting it from the decoded preceding B_1 and the decoded subsequent I frame, again followed by residue compression. In our previous work [10], we adopted an IBIB coding structure, which only has one intra-coding layer and one inter-coding layer. In the following subsections, we will elaborate the details of each coding layer of our new scheme.

2.1 Layer-1 Compression

Layer-1 adopts a convolutional neural network to compress the I frames. The network structure and parameters are specified in Fig. 2. We divide each I frame into patches of size 60×52. The network takes each patch as the input. In Fig. 2 $s = [s_h, s_w]$ represents the strides in height and width. We adopt different strides s to control the encoded feature dimension, followed by uniform scalar quantization and entropy coding, leading to different bit rates. Compared to our previous work [10], we improve this I frame compression network by adding a batch normalization (BN) [11] after each convolutional layer, to reduce internal covariate shift.

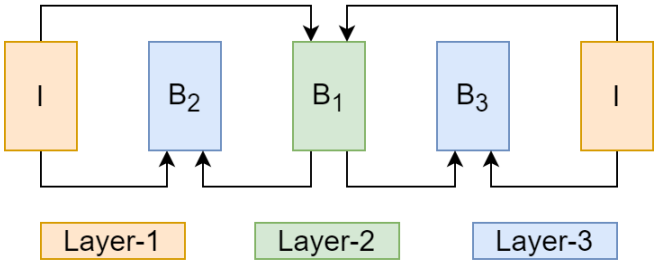


Figure 1: The proposed hierarchical coding structure with a GOP size of 4.

	Layer Type/Strides	(Filter Shape)×Number of Filters	Output Shape
	60×52×3 (Input)		
Encoder	Conv2D/s=[4,4]	(5×5) ×128	15×13×128
	BN	-	15×13×128
	ReLU	-	15×13×128
	Conv2D/s=[1,1]	(5×5) ×64	15×13×64
	BN	-	15×13×64
	ReLU	-	15×13×64
	Conv2D/s=[1,1]	(5×5) ×3	15×13×3
	BN	-	15×13×3
	ReLU	-	15×13×3
Uniform Scalar Quantization + Entropy Coding			
Decoder	Conv2DTranspose/s=[1,1]	(5×5) ×64	15×13×64
	BN	-	15×13×64
	ReLU	-	15×13×64
	Conv2DTranspose/s=[1,1]	(5×5) ×128	15×13×128
	BN	-	15×13×128
	ReLU	-	15×13×128
	Conv2DTranspose/s=[4,4]	(5×5) ×3	60×52×3
	ReLU	-	60×52×3

Figure 2: An illustration of the Layer-1 network.

2.2 Layer-2 Compression

Layer-2 compression is consisted of a bi-directional prediction network, and a residue coding network. It is worth noting that in [10], an IBIBI coding structure was adopted, hence the inter-frame (B frame) is predicted by adjacent previous and next decoded I frames. In contrast, this work adopts an $IB_2B_1B_3I$ structure, hence the Layer-2 inter-frame B_1 is predicted by the decoded reference blocks in the preceding and subsequent I frames. The prediction network layers are shown in Fig. 3. Branch 1 takes the preceding decoded I frame reference block as input, and branch 2 takes the subsequent decoded I frame reference block as input. To reduce the number of trainable parameters and to reduce the model size compared to the prediction net in [10], we adopt a smaller number of filters (8 filters) at the last layer of both two feature extraction branches. The outputs of both branches are then concatenated to be the input of the fusion block. Again, we add a BN after each convolution layer. The fusion block outputs the predicted B_1 patch $\widetilde{X}_{B_1}$. To compress the residue $X^r_{B_1} = X_{B_1} - \widetilde{X}_{B_1}$, we adopt the same network as the Layer-1 compression network, except that the network takes a residue patch of size $8 \times 8 \times 3$, while in Layer-1 compression, we use a larger patch size. The output of Layer-2 compression is the decoded B_1 patch residue $\widehat{X}^r_{B_1}$, which is then added to the predicted patch $\widetilde{X}_{B_1}$ to form the final decoded B_1 patch $\widehat{X}_{B_1} = \widetilde{X}_{B_1} + \widehat{X}^r_{B_1}$.

	Layer Type/Strides	(Filter Shape)×Number of Filters	Output Shape
	16×16×3 (Branch 1 Input); 16×16×3 (Branch 2 Input)		
Branch 1/Branch 2	Conv2D/s=[2,2]	(5×5) ×8	8×8×8
	BN+ReLU	-	8×8×8
	Conv2D/s=[1,1]	(5×5) ×16	8×8×16
	BN+ReLU	-	8×8×16
	Conv2D/s=[1,1]	(5×5) ×32	8×8×32
	BN+ReLU	-	8×8×32
	Conv2D/s=[1,1]	(5×5) ×64	8×8×64
	BN+ReLU	-	8×8×64
	Conv2D/s=[1,1]	(5×5) ×8	8×8×8
	BN+ReLU	-	8×8×8
Branch 1 and Branch 2 Concatenation			8×8×16
Fusion Block	Conv2D/s=[1,1]	(5×5) ×128	8×8×128
	BN+ReLU	-	8×8×128
	Conv2D/s=[1,1]	(5×5) ×256	8×8×256
	BN+ReLU	-	8×8×256
	Conv2D/s=[1,1]	(5×5) ×3	8×8×3
	ReLU	-	8×8×3

Figure 3: The proposed bi-directional prediction network.

2.3 Layer-3 Compression

Layer-3 compresses B_2 and B_3 frames. It is consisted of two prediction networks and two residue coding networks. Both prediction networks adopt the same structure as that in Fig. 3. The first prediction network takes the decoded $\widehat{X}_I$ and $\widehat{X}_{B_1}$ as the input of Branch 1 and Branch 2, respectively. It then outputs the predicted B_2 patch $\widetilde{X}_{B_2}$. The second prediction network takes the decoded $\widehat{X}_{B_1}$ and the subsequent $\widehat{X}_I$ as the input of Branch 1 and Branch 2, respectively. It then outputs the predicted B_3 patch $\widetilde{X}_{B_3}$.

Afterwards, the first residue coding network compresses the prediction residue $X^r_{B_2} = X_{B_2} - \widetilde{X}_{B_2}$, while the second residue coding network compresses the prediction residue $X^r_{B_3} = X_{B_3} - \widetilde{X}_{B_3}$. These two residue coding networks adopt the same structure as the Layer-2 residue coding network. Finally, the decoded residues $\widehat{X}^r_{B_2}$ and $\widehat{X}^r_{B_3}$ at the output of these two residue coding networks are added to the predicted patch to form the final decoded B_2 patch $\widehat{X}_{B_2} = \widetilde{X}_{B_2} + \widehat{X}^r_{B_2}$ and decoded B_3 patch $\widehat{X}_{B_3} = \widetilde{X}_{B_3} + \widehat{X}^r_{B_3}$, respectively.

3. EXPERIMENTAL STUDIES

In this section, we carry out experimental studies to demonstrate the effectiveness of our proposed model.

3.1 Datasets

We use three common test video sequences to evaluate the performance of our algorithm: *BlowingBubbles*, *BQSquare*, and *Johnny*. The resolutions of *BlowingBubbles* and *BQSquare* are both 240×416. The original resolution of *Johnny* is 720×1280. To save model training time, we resize *Johnny* to 240×416. We use the first 100 frames of each video sequence to train all the models, then we use the next 100 frames of each video sequence for testing. In total, we have 7 models to train: one I-frame compression model for Layer-1; one B_1-frame prediction model and one B_1-residue compression model in Layer-2; and finally, two prediction models and two residue compression models in Layer-3 to predict and compress B_2 and B_3. In terms of training complexity, it is higher than our previous method in [10], however, this new method enhances the coding efficiency.

3.2 Evaluation Metrics

We evaluate the performance of the proposed model by the multi-scale structural similarity (MS-SSIM) [12] versus the bit rates measured in bits per pixel (bpp) of the compressed video. The single-scale structural similarity (SSIM) was originally proposed in [13] to approximate the perceptual similarity between two images. Assume $\mathbf{x}$ is the ground-truth test frame and $\mathbf{y}$ is the decoded test frame. Let μ_x, σ_x^2 and σ_{xy} be the mean of $\mathbf{x}$, the variance of $\mathbf{x}$, and the covariance of $\mathbf{x}$ and $\mathbf{y}$, respectively. Approximately, μ_x and σ_x can be viewed as estimates of the luminance and contrast of $\mathbf{x}$, and σ_{xy} measures the tendency of $\mathbf{x}$ and $\mathbf{y}$ to vary together, thus an indication of structural similarity. In [13], the luminance, contrast and structure comparison measures between $\mathbf{x}$ and $\mathbf{y}$ were given as follows:

$$l(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{1}$$

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \tag{2}$$

$$s(\mathbf{x},\mathbf{y}) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \tag{3}$$

where C_1, C_2 and C_3 are small constants. Then the single-scale SSIM is defined as:

$$\text{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha} \cdot [c(\mathbf{x},\mathbf{y})]^{\beta} \cdot [s(\mathbf{x},\mathbf{y})]^{\gamma}, \tag{4}$$

where α, β and γ are parameters to define the relative importance of the three components.

To supply more flexibility than the single-scale SSIM, the MS-SSIM was proposed in [12] to incorporate the variations of viewing conditions. In particular, it is obtained by combining the SSIM at different image scales using

$$\text{MS-SSIM}(\mathbf{x},\mathbf{y}) = [l_M(\mathbf{x},\mathbf{y})]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(\mathbf{x},\mathbf{y})]^{\beta_j} \cdot [s_j(\mathbf{x},\mathbf{y})]^{\gamma_j}, \tag{5}$$

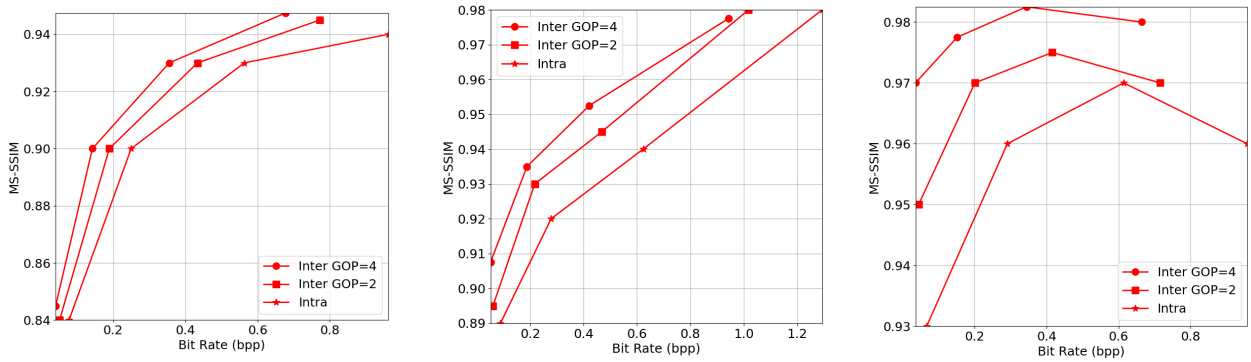
where j is the scale index, M is the number of scales adopted, and similar to (4), the exponents α_M, β_j and γ_j are used to adjust the relative importance of different components.

3.3 Objective Analysis

We quantitatively analyze the rate-distortion performance of our proposed hierarchical video compression model in Fig. 4. Both the MS-SSIM and bpp values are averaged over 100 test frames for each video. For all three video sequences, we observe that the proposed Inter GOP = 4 can achieve the same level of MS-SSIM with less bit rates than Inter GOP = 2 [10] and the baseline Intra-frame coding.

In particular, Fig. 4 (a) shows that for *BlowingBubbles*, at MS-SSIM 0.900, the baseline Intra-frame coding method requires 0.25 bpp, while Inter GOP = 2 and the proposed Inter GOP = 4 requires 0.19 bpp and 0.14 bpp, which reduces the bit rates by 24% and 44% respectively, compared to the baseline. At MS-SSIM 0.920, the baseline requires 0.46 bpp, while Inter GOP = 2 and the proposed Inter GOP = 4 requires 0.35 bpp and 0.28 bpp, which reduces the bit rates by 24% and 39% respectively. At MS-SSIM 0.940, the baseline requires 0.96 bpp, while Inter GOP = 2 and the proposed Inter GOP = 4 requires 0.66 bpp and 0.54 bpp, which reduces the bit rates by 31% and 44% respectively. Similar results can be obtained in Fig. 4 (b) for *BQSquare*.

Besides, for *Johnny*, we observe from Fig. 4 (c) that the proposed Inter GOP = 4 achieves an MS-SSIM of 0.970 at 0.04 bpp, but Inter GOP = 2 and baseline achieve this level of MS-SSIM at much higher bit rates, 0.2 bpp and 0.61 bpp, respectively. Further, the proposed Inter GOP = 4 achieves its highest MS-SSIM 0.983 at 0.34 bpp, while Inter GOP = 2 achieves its highest MS-SSIM 0.975 at 0.41 bpp, and the baseline achieves its highest MS-SSIM 0.970 at 0.61 bpp. We observe a performance loss at the highest bpp values in Fig. 4 (c), for all three models. It is because a large area of *Johnny* scene is very smooth, but we adopted uneven strides in height and width $s = [2, 4]$ to obtain these bit rates in the I-frame and residue-frame encoding and decoding layers. This severely dampened the structural quality of the decoded I frames, which further affected the inter-frames.



(a) (b) (c)

Figure 4: The rate-distortion curves of (a) *BlowingBubbles*, (b) *BQSquare*, and (c) *Johnny*.

Ground-Truth	I	B	B_1	B_2/B_3
Inter GOP=4	25%	0%	25%	50%
Inter GOP=2	50%	50%	0%	0%
Intra	100%	0%	0%	0%

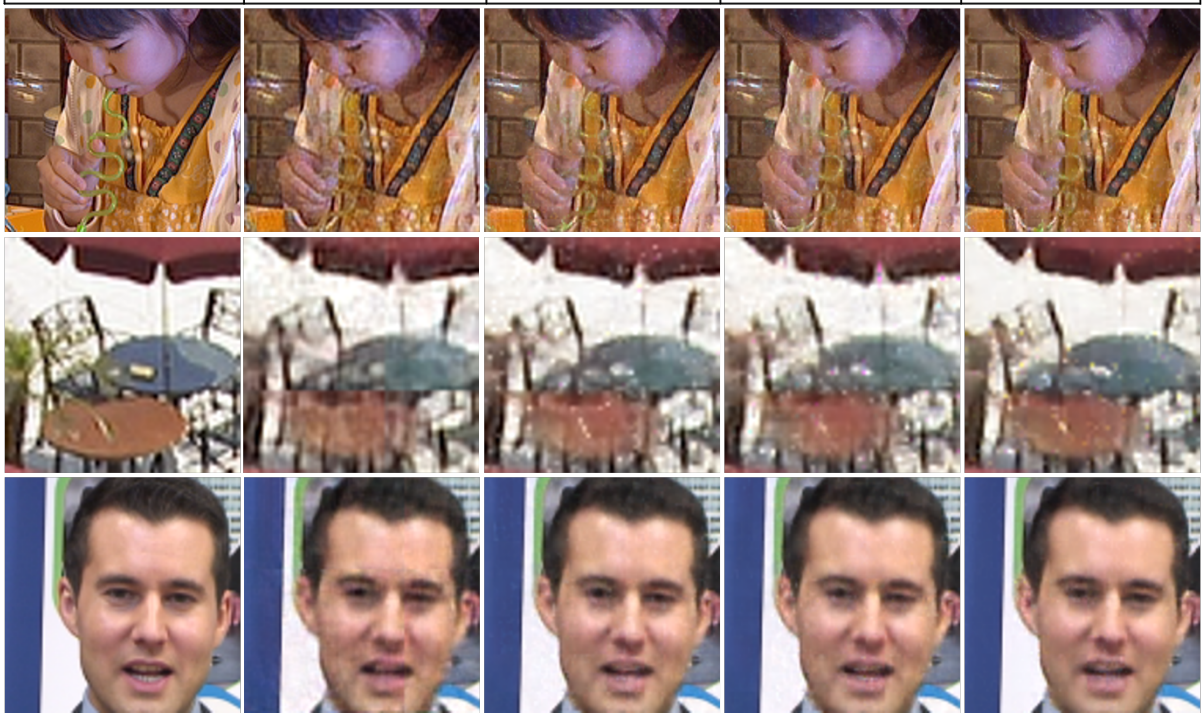


Figure 5: The enlarged ground-truth frame and the decodings of I, B, B_1, and B_2/B_3 frames. The B frame refers to the inter-frame in Inter GOP=2, in which the coding structure is IBIBI. The three rows of images from top to bottom in this figure are *BlowingBubbles*, *BQSquare*, and *Johnny*, respectively.

3.4 Subjective Analysis

We also display the decoded frames with three different coding structures in Fig. 5. The table at the top describes the percentage of each frame type in a different coding structure. The Intra-frame coding has 100% I frames that are compressed independently. The Inter GOP = 2 coding has an IBIBI coding structure, in which 50% are I frames and another 50% are B frames, with each B frame predicted by the adjacent previous and next decoded I frames. In contrast, our proposed Inter GOP = 4 has an $IB_2B_1B_3I$ structure, in which 25% are I frames, 25% are B_1 frames, and the remaining 50% are B_2 or B_3 frames.

We observe from Fig. 5 that the decoded I frames have the worst visual quality. They are blurry (such as the curly straw in *BlowingBubbles*), noisy (such as the white area in *BQSquare*), or have blockiness artifacts (the table area in *BQSquare* and the chin area of the man in *Johnny*). Our proposed Inter GOP= 4 scheme has the smallest percentage (25%) of such I frames, leading to a higher average MS-SSIM. However, the Inter GOP= 2 and Intra coding schemes have 50% and 100% such low-quality I frames. On the other hand, the decoded B frame, B_1 frame, and B_2/B_3 frames have very close visual quality. Since our proposed Inter GOP= 4 is mainly consisted of these type of frames, the overall visual quality of our scheme is better than the other two methods in comparison.

4. CONCLUSIONS

In this work, we proposed a patch-wise hierarchical CNN for video coding. The coding structure has three layers. While Layer-1 performs intra-frame coding, Layer-2 and Layer-3 both perform bi-directional predictive coding. This scheme significantly enhances the coding efficiency compared to intra-frame coding and inter-frame coding that has only one prediction and residue coding layer. The superiority of our new model is demonstrated by experimental studies on common test video sequences.

References

[1] Yang, R., Mentzer, F., Gool, L. V., and Timofte, R., "Learning for video compression with hierarchical quality and recurrent enhancement," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 6628–6637 (2020).

[2] Yang, R., Mentzer, F., Van Gool, L., and Timofte, R., "Learning for video compression with recurrent auto-encoder and recurrent probability model," *IEEE Journal of Selected Topics in Signal Processing* (2020).

[3] Habibian, A., Rozendaal, T. v., Tomczak, J. M., and Cohen, T. S., "Video compression with rate-distortion autoencoders," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 7033–7042 (2019).

[4] Wu, C.-Y., Singhal, N., and Krahenbuhl, P., "Video compression through image interpolation," in [*Proceedings of the European Conference on Computer Vision (ECCV)*], 416–431 (2018).

[5] Liu, H., Shen, H., Huang, L., Lu, M., Chen, T., and Ma, Z., "Learned video compression via joint spatial-temporal correlation exploration," in [*Proceedings of the AAAI Conference on Artificial Intelligence*], **34**(07), 11580–11587 (2020).

[6] Mao, J. and Yu, L., "Convolutional neural network based bi-prediction utilizing spatial and temporal information in video coding," *IEEE Transactions on Circuits and Systems for Video Technology* **30**(7), 1856–1870 (2019).

[7] Choi, H. and Bajić, I. V., "Deep frame prediction for video coding," *IEEE Transactions on Circuits and Systems for Video Technology* **30**(7), 1843–1855 (2019).

[8] Yan, N., Liu, D., Li, H., Li, B., Li, L., and Wu, F., "Convolutional neural network-based fractional-pixel motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology* **29**(3), 840–853 (2018).

[9] Liu, K., Liu, D., Li, H., and Wu, F., "Convolutional neural network-based residue super-resolution for video coding," in [*2018 IEEE Visual Communications and Image Processing (VCIP)*], 1–4, IEEE (2018).

[10] Khan, R. and Liu, Y., "Motion-aware deep video coding network," in [*Big Data II: Learning, Analytics, and Applications*], **11395**, 113950B, International Society for Optics and Photonics (2020).

[11] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in [*International conference on machine learning*], 448–456, PMLR (2015).

[12] Wang, Z., Simoncelli, E. P., and Bovik, A. C., "Multiscale structural similarity for image quality assessment," in [*The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*], **2**, 1398–1402, Ieee (2003).

[13] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing* **13**(4), 600–612 (2004).