Learning-Based Scalable Video Coding with Spatial and Temporal Prediction

Martin Benjak[1], Yi-Hsin Chen[2], Wen-Hsiao Peng[2], Jörn Ostermann[1]
[1]Institut für Informationsverarbeitung, Leibniz Universität Hannover, Germany
[2]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
benjak@tnt.uni-hannover.de

***Abstract*—In this work, we propose a hybrid learning-based method for layered spatial scalability. Our framework consists of a base layer (BL), which encodes a spatially downsampled representation of the input video using Versatile Video Coding (VVC), and a learning-based enhancement layer (EL), which conditionally encodes the original video signal. The EL is conditioned by two fused prediction signals: a spatial inter-layer prediction signal, that is generated by spatially upsampling the output of the BL using super-resolution, and a temporal inter-frame prediction signal, that is generated by decoder-side motion compensation without signaling any motion vectors. We show that our method outperforms LCEVC and has comparable performance to full-resolution VVC for high-resolution content, while still offering scalability.**

***Index Terms*—VVC, video coding, spatial scalability, scalable coding, conditional coding**

I. Introduction

Delivering video content over the Internet is very energy and bandwidth intensive. In order to keep the required bandwidth and thus also the energy consumption as low as possible, adaptive streaming standards such as MPEG-DASH [1] and HLS [2] are used. MPEG-DASH and HLS allow for providing different representations of a media file, e.g. different resolutions and qualities. The user device can adaptively switch between these representations, providing a high quality experience while minimizing the bandwidth. In a setting such as video-on-demand, where one video is encoded once, but transmitted many times, independently decodable representations offer the lowest energy footprint. However, in a broadcast setting, where all streams are transmitted simultaneously, having interdependent representations results in the lowest overall bandwidth. An example of such a setting is the planned deployment of Low Complexity Enhancement Video Coding (LCEVC) by the Brazilian SBTVD Forum [3].

Scalable video coding describes methods that provide multiple interdependent representations of the same video, varying in spatial, temporal or fidelity dimensions. The architecture of a scalable codec consists of a base layer (BL) and one or more enhancement layers (EL). The BL representation is encoded with a standard codec and the EL representations are encoded with a scalable video codec that encodes only the difference between the BL and EL representations.

Scalable video coding extensions have been developed for all recent video coding standards. Advanced Video Coding (AVC) has been extended by Scalable Video Coding (SVC) [4], High Efficiency Video Coding (HEVC) [5] has been extended by the HEVC Scalability Extension (SHVC) [6] and the latest video coding standard Versatile Video Coding (VVC) has built-in support for multi-layer coding which also provides scalability [7]. Unlike the aforementioned coding standard-specific scalable extensions, LCEVC is BL codec agnostic [8]. However, these scalable video coding extensions, other than LCEVC, have never found many real-world applications.

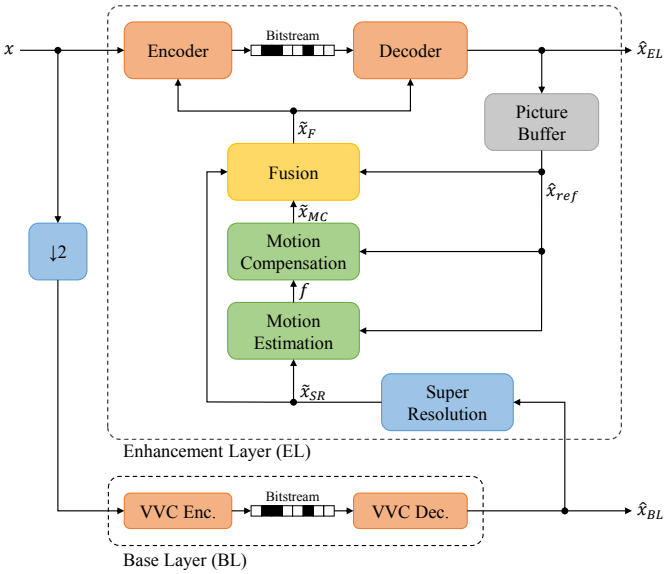

Fig. 1: Architecture of our proposed hybrid layered scalable codec with spatial inter-layer and temporal inter-frame prediction in the EL (inter codec).

In recent years, several learned image [9]–[12] and video compression [13]–[20] methods have been proposed. The learning-based video codec by Li et al. [20] is even able to outperform the prototype of the next-generation video coding standard [21]. Besides these end-to-end learned codecs, there are also hybrid approaches, that combine conventional codecs with learning-based modules. Examples for learning-based modules in hybrid codecs are loop filters [22], intra [23] and inter prediction modes [24]. A special form of hybrid codecs are learning-based scalable coding extensions, that use conventional codecs as BL and encode the high-resolution details using learning-based codecs [25]–[28]. However, these

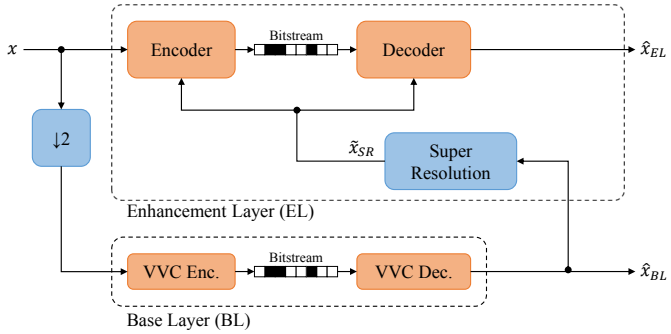

Fig. 2: Architecture of our proposed hybrid layered scalable codec with spatial inter-layer prediction in the EL (intra codec).

learning-based scalable extensions are all designed for the compression of images or I-frames.

In this paper, we present a hybrid layered approach for scalable video coding based on CAESR [28]. While CAESR only uses the downscaled representation of the input image from the BL codec as a prediction signal, we additionally propose to use motion-compensation to improve the prediction signal, without signaling any additional information such as motion vectors or flow-maps. Furthermore, we upscale the BL output using super-resolution before feeding it to the EL codec, while CAESR uses bicubic upscaling and applies super-resolution only in a final step after decoding the EL. A deep conditional [17] autoencoder is serving as the codec of the EL. During downscaling and quantization in the BL, mainly high-frequency details are lost. While super-resolution can recover some of the lost detail, motion-compensation has the benefit that it can access higher quality reference pictures to recover additional detail. By leveraging both, spatial inter-layer prediction and temporal high-resolution inter-frame prediction, we aim to achieve a compression performance that is comparable to full-resolution VVC in low-delay configuration, while providing the flexibility of scalable coding.

II. Proposed Solution

The general structure of our proposed hybrid layered scalable codec is as follows. The input picture x is bicubically downscaled and encoded with a standardard codec, which forms the BL. In this work, we choose VVC as the codec for the BL, but our method is generally BL codec agnostic. The reconstructed BL output $\hat{x}_{BL}$ is then upscaled using a neural super-resolution network. Due to its low complexity and good performance, we choose EDSR [29] for this task. The super-resolution upscaled signal $\tilde{x}_{SR}$ is then fed together with the input picture x into the EL.

Like other video codecs, our EL has two modes: one for I-frames and one for P-frames. The I-frame (intra) mode relies solely on inter-layer prediction using the super-resolution signal $\tilde{x}_{SR}$. The P-frame (inter) mode implements both, inter-layer prediction using the super-resolution signal $\tilde{x}_{SR}$ and

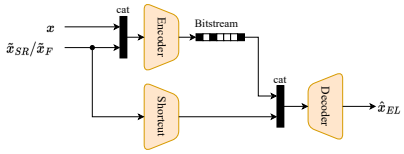

Fig. 3: Internal structure of the conditional autoencoder.

motion-compensated inter-frame prediction. In the following, we will first describe the intra-mode and then the inter-mode.

In intra-mode, the EL is implemented as shown in Fig. 2. The encoder and decoder are based on the end-to-end learned image compression model from [12]. To enable conditional coding, we extend their model with a scheme similar to [17]: the input signal x and the spatially predicted signal $\tilde{x}_{SR}$ are concatenated and fed into the encoder. Since the latent space dimensionality differs from the predicted signal $\tilde{x}_{SR}$, the two cannot be concatenated directly. Instead, a shortcut network is used, which has the same architecture as the encoder network (see Fig. 3). $\tilde{x}_{SR}$ is fed into the shortcut and the output is concatenated together with the quantized latent.

In inter-mode, the EL is implemented as shown in Fig. 1. We use the same conditional autoencoder architecture as in our intra-mode. However, the prediction signal used as the condition for the autoencoder is calculated in a different way: We still obtain the spatial inter-layer prediction signal $\tilde{x}_{SR}$ in the same way as described above. To exploit remaining redundancies between successive frames, we additionally introduce the motion-compensated prediction signal $\tilde{x}_{MC}$. Both prediction signals $\tilde{x}_{MC}$ and $\tilde{x}_{SR}$ are fused together with $\hat{x}_{ref}$ using a fusion network into the fused prediction $\tilde{x}_F$. For the fusion network, we use the same U-Net structure as in [19].

Motion estimation generates the flow-map f between $\tilde{x}_{SR}$ and the past frame in the picture buffer $\hat{x}_{ref}$. In our work, we use the PWC-Net [30] as the motion estimation network. Using $\tilde{x}_{SR}$ instead of x to estimate f has the advantage that the estimation can be done on the decoder side. Therefore, no additional information needs to be signaled. Finally, the flow-map is used to warp the reference picture $\hat{x}_{ref}$ into a prediction for x. In Section IV, we will show the benefits of decoder-side motion estimation.

Our inter-mode differs from other codecs' inter-modes in that we combine both spatial and temporal prediction. In some cases this combination of both modes may not be optimal, so we introduce a Rate Distortion Optimization (RDO) step during encoding: All potentially inter-predicted frames (P-frames) are encoded twice. Once using the intra-mode and once using the inter-mode. The best mode is signaled.

III. Implementation and Training

We train our models using the BVI-DVC dataset [31]. The dataset consists of 200 UHD (3840 × 2160) sequences of 64 frames each. Additionally, the dataset also contains downscaled representations of these 200 sequences with resolutions of 1920 × 1080, 960 × 540, and 480 × 270 pixels. First, all 800 sequences of the dataset are bicubically downscaled by a factor

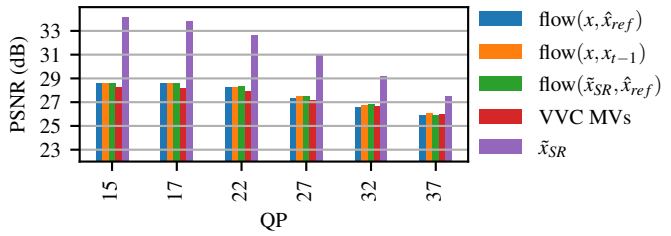

Fig. 4: Average RGB-PSNR for reference pictures warped using different flow-maps over BL QPs in the JVET dataset classes A, B, C, D. Reference pictures were generated using the corresponding intra EL codecs.

of 2 and then encoded using the VVC [7] reference software VTM version 19.0 in low-delay P configuration (YUV 4:2:0) for the quantization parameters (QP) 15, 17, 22, 27, 32, and 37. After this procedure, the reconstructed frames are used as our reconstructed BL representation $\hat{x}_{BL}$. Together with their corresponding unscaled and uncoded original frames x, they form our dataset. This dataset is then randomly split into 1000 frames for validation and the rest for training.

We optimize our model using adam and the RD loss

$$\mathcal{L}(\lambda) = D(\hat{x}_{EL}, x) + \lambda R. \tag{1}$$

The distortion D is measured by Mean Squared Error (MSE) between the original frames x and $\hat{x}_{EL}$. The rate R is calculated using the Shannon entropy of the latent space. With λ, the trade-off between distortion and rate can be configured. This loss function is commonly used in the field of end-to-end learned image and video compression.

Before training, the training data is converted from YUV color space to RGB. In each epoch, we randomly crop one co-aligned 256 x 256 patch from x and one 128 x 128 patch from $\hat{x}_{BL}$ per input datum. One model is trained per BL QP and per λ. We choose a batch size of 22 and an initial learning rate of 10^{-4}. The parameters of the motion estimation and super-resolution networks are preloaded with the weights provided by their respective authors. For the conditional autoencoder, which consists of an encoder, a decoder and a shortcut network, we use the implementation provided by CompressAI [32]. All three modules of the conditional autoencoder are preloaded with the pre-trained model at quality level 2 provided by CompressAI.

The training is done in several steps: First, only the fusion network is trained using a MSE loss function for 10 epochs. Then, only the conditional autoencoder is trained for 8 epochs, while the fusion network is kept static. After that, the fusion network and the conditional autoencoder are trained jointly for another 5 epochs. Finally, all models, including motion estimation and super-resolution, are trained jointly for another 16 epochs. During the last 6 epochs, the learning rate alternates between $1/2$ and $1/4$ of its initial value. In total, the final model is trained for 39 epochs. After this training procedure, the model is trained for another 50 short epochs with only 264 training image pairs each. Between each of these short epochs, the model is evaluated using the validation dataset. Out of these 50 short epochs, the model state with the best evaluation performance is kept as the final model.

The reference picture for the inter-mode is generated during the first 28 epochs by encoding the past frame using an already trained intra codec. After epoch 28, the reference picture is generated by using the pre-trained intra codec followed by two iterations of the current state of the inter codec.

IV. Selection of Flow Estimation Methods

Conventionally, motion estimation is done by estimating the motion between the reference picture and the original frame. Since the original frame is unknown to the decoder, the resulting flow-map must be signaled. In this work, however, we can also use the spatial inter-layer prediction $\tilde{x}_{SR}$ or the upscaled motion vectors (MVs) that VVC used to encode the BL, both of which are already known to the decoder. Besides these three methods, the flow-map between the original frame x and the past frame x_{-1} could also be estimated. We compare all four methods by warping the reference picture with the flow-maps estimated by each method and measuring the PSNR between these warped frames and the original frames. The flow-maps are not coded and the reference pictures are generated by feeding the past frame through a trained intra-mode EL codec.

The results of this experiment using the JVET test sequences [33] are shown in Fig. 4. It can be seen that the VVC MVs perform the worst for most QPs while the other methods give comparable results to each other. It can also be seen, that $\tilde{x}_{SR}$ outperforms all forms of motion compensation. From this it can be concluded that motion compensation alone, without spatial prediction, is not suitable. Since motion estimation between $\tilde{x}_{SR}$ and $\hat{x}_{ref}$ has the additional advantage that nothing needs to be signaled, this is clearly the best choice to enhance the already very high performing spatial prediction.

V. Results and Discussion

We compare our method with 3 state-of-the-art methods for spatially scalable video coding: LCEVC, VVC multi-layer and CAESR. All 3 reference methods use the same bicubic downsampling filter to generate the BL input as our approach and operate in YUV color space with 4:2:0 chroma subsampling. LCEVC and CAESR share the same BL configuration with our method (low-delay P @ QP 15, 17, 22, 27, 32, 37). For VVC multi-layer, we use BL and EL configurations based on [34] with the only difference being the downsampling filter and the use of low-delay P configuration instead of random access. The VVC multi-layer EL can perform both intra-layer and inter-layer predictions. The two step width parameters of LCEVC are calculated according to Annex 1 of [35]. The λ parameter of CAESR is selected based on a grid search.

We evaluate our intra and inter codecs as well as our 3 reference methods using the JVET test sequences [33]. All sequences are first encoded in the BL in YUV 4:2:0, converted to RGB, and finally encoded with the corresponding EL codec. The EL λ is selected manually to optimize the BD-Rate. The selected λ/QP pairs are 0.0005/37, 0.001/32, 0.002/27,

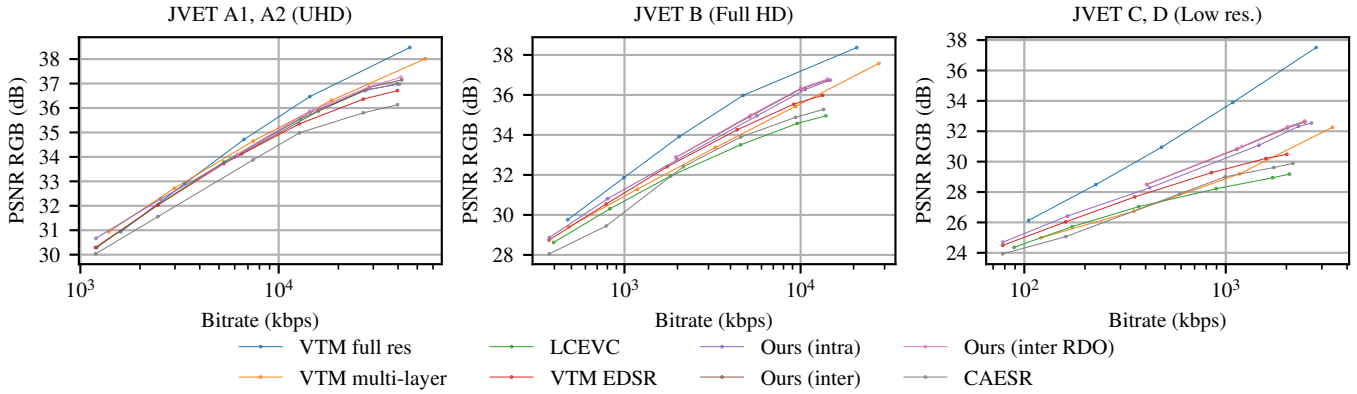

Fig. 5: Comparison between our proposed method with VTM [7] in full resolution, VTM with multi-layer configuration, VTM in half resolution and then upsampled using EDSR [29], LCEVC [8] and CAESR [28].

0.0075/22, 0.011/17 and 0.013/15 for intra and 0.0025/27, 0.008/22, 0.011/17 and 0.015/15 for inter. The inter codec has an intra period of 8. The inter codec with RDO has no fixed intra period, instead the frame type is determined using Equation 1 and signaled for each frame. To simulate the bitrate required to signal the frame type, we add one byte to the bitrate per frame. Fig. 5 shows the results broken down into separate classes of the JVET test sequences. Classes A1 and A2 contain a resolution of 3840 x 2160 pixels, class B 1920 x 1080 pixels, class C 832 x 480 pixels, and class D 416 x 240 pixels. Table I lists the BD-rates for all three configurations of our codec. The inter codec is unstable at low bitrates due to the extremely low λ setting. High but stable λ values result in a poor RD-performance, comparable to CAESR. To ensure a fair comparison over the entire bitrate range, we have replaced the missing RD points of our inter and inter RDO codecs with the intra codec points for BD-rate calculation.

Our intra codec outperforms full-resolution VVC at low bitrates for classes A1 and A2. The inter codec shows gains over the intra codec for classes B, C, and D, while the intra codec is better suited for classes A1 and A2. The combination of inter and intra codecs using RDO gives the best results for all classes. RDO selected 55.20 %, 68.53 %, 71.86 %, 73.09 %, and 74.63 % of all frames to be P-frames for classes A1, A2, B, C, and D, respectively. In contrast, a fixed intra period of 8 results in 87.50 % of all frames being P-frames.

LCEVC shows similar performance to our approach for classes A1 and A2, while we outperform it for classes B, C, and D. VTM (VVC) with a multi-layer configuration outperforms our approach for UHD content, while our approach outperforms it for lower resolutions. Although CAESR was trained in the same way as our approach, it performs poorly when evaluated with a BL in a low-delay configuration. However, for all-intra, we were able to reproduce the results reported in [28]. The reason for CAESR's poor performance is its less steep RD-curve. If we increase its λ, the PSNR increases, but the overall RD-performance is worse due to the high bitrate increase.

Table II shows a complexity comparison of our models with CAESR in terms of multiply-accumulate (MAC) operations used to encode and decode a frame, as well as the number of trainable parameters. Note that our inter model has to decode during the encoding time to reconstruct reference pictures. CAESR and our intra model do not need to do so.

TABLE I: BD-rates (RGB) relative to VTM 19.0 in low resolution and upsampled using EDSR. Negative BD-rates indicate increased coding efficiency.

Codec	Classes A1, A2	Class B	Classes C, D
Ours (intra)	-9.48 %	-8.74 %	-17.99 %
Ours (inter)	-8.08 %	-12.59 %	-23.72 %
Ours (inter RDO)	-11.14 %	-13.20 %	-24.05 %
multi-layer VVC	-14.82 %	11.15 %	43.71 %
LCEVC	-5.67 %	35.07 %	52.21 %
CAESR	28.16 %	39.01 %	53.50 %

TABLE II: Complexity comparison

Model	Size	Encode MACs	Decode MACs
Ours (intra)	37M	0.77 M/px	1.38 M/px
Ours (inter)	47M	2.21 M/px	1.80 M/px
CAESR	28M	0.4 M/px	1.28 M/px

VI. Conclusion

We present a hybrid approach for spatial scalability consisting of a BL, which encodes a spatially downsampled representation of the the input video using VVC, and a learning-based EL. The EL combines both spatial inter-layer prediction and temporal inter-frame prediction into a fused prediction signal for a conditional autoencoder-based coding scheme. For the inter-layer prediction, we use a super-resolution network that upscales the BL representation by a factor of 2. The inter-frame prediction is done by motion estimation and compensation using the inter-layer prediction signal without signaling any motion information. In addition, we use RDO to determine whether inter-frame prediction should be used on a frame-by-frame basis. Our approach provides similar performance to full-resolution VVC for UHD content, while still offering scalability. For lower resolution content, we can show that inter-frame prediction improves the EL performance.

REFERENCES

[1] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE multimedia*, vol. 18, no. 4, pp. 62–67, 2011.

[2] R. Pantos, "Http live streaming," Internet Requests for Comments, RFC Editor, RFC 8216, August 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8216.txt

[3] T. Biatek, M. Abdoli, M. Raulet, A. Wieckowski, C. Lehmann, B. Bross, P. De Lagrange, E. François, R. Schaefer, and J. Lefeuvre, "Versatile video coding for 3.0 next generation digital tv in brazil," *SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING*, vol. 7, 2021.

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable h. 264/mpeg4-avc extension," in *2006 International Conference on Image Processing*. IEEE, 2006, pp. 161–164.

[5] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, 2015.

[7] 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.

[8] G. Meardi, S. Ferrara, L. Ciccarelli, G. Cobianchi, S. Poularakis, F. Maurer, S. Battista, and A. Byagowi, "Mpeg-5 part 2: Low complexity enhancement video coding (lcevc): Overview and performance evaluation," *Applications of Digital Image Processing XLIII*, vol. 11510, pp. 238–257, 2020.

[9] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.

[10] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[11] 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.

[12] 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.

[13] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 006–11 015.

[14] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8503–8512.

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

[16] 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 Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3182–3196, 2020.

[17] T. Ladune, P. Philippe, W. Hamidouche, L. Zhang, and O. Déforges, "Optical flow and mode selection for learning-based video coding," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[18] Z. Hu, G. Lu, and D. Xu, "Fvc: A new framework towards deep video compression in feature space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1502–1511.

[19] Y.-H. Ho, C.-P. Chang, P.-Y. Chen, A. Gnutti, and W.-H. Peng, "Canf-vc: Conditional augmented normalizing flows for video compression," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*. Springer, 2022, pp. 207–223.

[20] J. Li, B. Li, and Y. Lu, "Neural video compression with diverse contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 616–22 626.

[21] M. Coban, R.-L. Liao, K. Naser, J. Ström, and L. Zhang, "Algorithm description of enhanced compression model 9 (ecm 9)," *ITU-T and ISO/IEC JVET-AD2025*, 2023.

[22] W.-S. Park and M. Kim, "Cnn-based in-loop filtering for coding efficiency improvement," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE, 2016, pp. 1–5.

[23] Z. Pan, P. Zhang, B. Peng, N. Ling, and J. Lei, "A cnn-based fast inter coding method for vvc," *IEEE Signal Processing Letters*, vol. 28, pp. 1260–1264, 2021.

[24] M. Benjak, H. Meuel, T. Laude, and J. Ostermann, "Enhanced machine learning-based inter coding for vvc," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2021, pp. 021–025.

[25] Y.-H. Tsai, M.-Y. Liu, D. Sun, M.-H. Yang, and J. Kautz, "Learning binary residual representations for domain-specific video streaming," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[26] M. Akbari, J. Liang, and J. Han, "Dsslic: Deep semantic segmentation-based layered image compression," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2042–2046.

[27] W.-C. Lee, C.-P. Chang, W.-H. Peng, and H.-M. Hang, "A hybrid layered image compressor with deep-learning technique," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[28] C. Bonnineau, W. Hamidouche, J.-F. Travers, N. Sidaty, J.-Y. Aubié, and O. Deforges, "Caesr: Conditional autoencoder and super-resolution for learned spatial scalability," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2021, pp. 1–5.

[29] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[30] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.

[31] D. Ma, F. Zhang, and D. R. Bull, "Bvi-dvc: A training database for deep video compression," *IEEE Transactions on Multimedia*, vol. 24, pp. 3847–3858, 2021.

[32] 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.

[33] E. Alshina, R.-L. Liao, S. Liu, and A. Segall, "Common test conditions and evaluation procedures for neural network-based video coding technology," *ITU-T and ISO/IEC JVET-AD2016*, 2023.

[34] S. Iwamura, P. de Lagrange, and M. Wien, "Verification test plan for vvc multilayer coding," *ITU-T and ISO/IEC JVET-AD2021*, 2023.

[35] F. Maurer, L. Ciccarelli, and S. Ferrara, "[lcevc] verification test – overview of bitrate, psnr, vmaf, and md5 checksums," *ISO/IEC JTC1/SC29/WG4 m56330*, 2021.