Spatial-Temporal Autoencoder with Attention Network for Video Compression

Neetu Sigger[1[0000-0002-9275-5691]], Naseer Al-Jawed[1[0000-0002-4585-6385]], Tuan Nguyen[2[0000-0003-0055-8218]]

[1] The University of Buckingham, Buckingham, United Kingdom
{neetu.sigger, naseeraljawed}@buckingham.ac.uk
[2] University of Greenwich, London, United Kingdom
{tuan.nguyen}@greenwich.ac.uk

Abstract. Deep learning-based approaches are now state of the art in numerous tasks, including video compression, and are having a revolutionary influence in video processing. Recently, learned video compression methods exhibit a fast development trend with promising results. In this paper, taking advantage of the powerful non-linear representation ability of neural networks, we replace each standard component of video compression with a neural network. We propose a spatial-temporal video compression network (STVC) using the spatial-temporal priors with an attention module (STPA). On the one hand, joint spatial-temporal priors are used for generating latent representations and reconstructing compressed outputs because efficient temporal and spatial information representation plays a crucial role in video coding. On the other hand, we also added an efficient and effective Attention module such that the model pays more effort on restoring the artifact-rich areas. Moreover, we formalize the rate-distortion optimization into a single loss function, in which the network learns to leverage the Spatial-temporal redundancy presented in the frames and decreases the bit rate while maintaining visual quality in the decoded frames. The experiment results show that our approach delivers the state-of-the-art learning video compression performance in terms of MS-SSIM and PSNR.

Keywords: Video Compression, Deep Learning, Auto-Encoder, Rate-Distortion Optimization, Attention Mechanism.

1 Introduction

There have been high demands on video compression over recent years, with the efficient transmission of top resolution and high-quality video data over the bandwidth-limited Internet. Especially during the COVID pandemic, the increasing data traffic was used for online classes, virtual meetings, Netflix, YouTube, online gaming, etc. As a result, there's a growing demand for more practical video compression schemes to speed up the method of exchanging visual media over the bandwidth-limited Internet. By

utilizing redundancies within the data to produce a more miniature representation, compression methods are employed to handle the expanding sizes of stored media.
Deep Learning (DL) is revolutionizing image and video processing, and DL-based techniques are state-of-the-art in several related problems such as classification, detection, or compression [1], [2].

In the past few decades, several video compression algorithms have been standardized, e.g., MPEG [3], H.264 [4], and H.265 [5]. These standards are hand-made, and the components in the compression framework cannot be optimized together. Motivated by the success of the deep neural networks (DNN) in improving image compression rate-distortion performance [6], [7] various DNN based video compression architectures [8], [9] have been developed. In these learned video compression methods, the whole network is optimized in an end-to-end manner. The DL-based image compression relies on the ability of DNN to extract meaningful representations of two-dimensional data because the latent space of an image represented by the network must contain information about the most important features and structures in the image. The convolutional auto-encoder is particularly suitable for image processing because it can take advantage of the spatial redundancy in the image.

It is still a difficult task to figure out how to generate and compress motion information that is optimized for video compression. To decrease the spatial-temporal redundancy in video sequences, video compression algorithms mainly rely on optical flow information. Furthermore, developing a DL-based video compression system that minimizes the rate-distortion-based objective for both residual and motion information is also another challenge. Recently, image compression algorithms [10], [11] based on machine learning methods have shown great superiority in coding efficiency for spatial redundancy removal compared with conventional codecs. These models get benefit from non-linear transforms, DNN based conditional entropy model, and a joint rate-distortion optimization (RDO) under an end-to-end learning strategy. Learned video compression can be extended from the image compression approach by further exploiting the temporal redundancy or correlation. Besides, the artifacts in frames also harm the performance of video-oriented tasks (e.g., video summarization [12], action recognition, and localization [13]). Accordingly, artifact reduction in frames, which aims to reduce the artifacts and recover missed details from the frame and pay more attention to complex regions to improve coding performance, becomes a hot topic in the multimedia field [14].

Therefore, this paper proposes a spatial-temporal video compression network (STVC) using the spatial-temporal priors with an attention module (STPA). As shown in Fig. 1, the proposed STVC approach uses convolutional networks for representing inputs, reconstructing compressed outputs. Specifically, the proposed STPA network contains an attention network in both the encoder and decoder. Given a sequence of inputs $\{x_1, x_2 \dots \dots x_t\}$, the encoder of STPA generates the latent representations $\{y_1, y_2 \dots \dots y_t\}$, and the decoder also reconstructs the compressed outputs $\{\hat{x}_1, \hat{x}_2 \dots \dots x_t\}$ from $\{y_1, y_2 \dots \dots y_t\}$. Besides, we also add an attention module into STPA network architecture. Attention modules can make learned models pay more attention to complex regions to improve our coding performance. The attention mechanism is also embedded to generate a more compact representation for both latent features and

hyperpriors. Our attention model is applied at different layers (not only for quantized features at the bottleneck), to adapt intelligently through the end-to-end learning framework.

Our main aim is to improve the flaws of the traditional video compression methods by replacing each traditional aspect with its minimalistic neural network equivalent. The following is a summary of the contributions:

1. We propose a spatial-temporal video compression network (STVC) that jointly learns motion compression, motion estimation, and residual compression. It optimizes all the components simultaneously under the scrutiny of a single loss feature.
2. We propose the spatial-temporal priors with an attention module (STPA). To compress the corresponding motion and residual to consider existing spatial and temporal data and compress them in a quantized latent representation using autoencoders, multi-scale connections, and convolutions. We also apply the attention module together with the autoencoder to reduce the artifacts and recover missing details from the frame and pay more attention to complex regions.

Our experiments validate that compressing the motion information using our approach can significantly improve the compression performance. Our framework outperforms the DVC [8], H.264 [4], and H.265 [5] when measured by MS-SSIM and PSNR. In the following, Section 2 presents the related works. The proposed STVC and STPA are introduced in Section 3. Then, the experiments in Section 4 validate the performance of the proposed STVC approach to the existing learned video compression approaches. Finally, concluding remarks and future works are described in Section 5.

2 Related Work

In this section, we discuss several deep learning methods related to the image and video compressions that are highly relevant to our work.

2.1 Learned Image Compression

DNN-based image compression methods are generally based on automatic encoders. For the first time [15], it is proposed to use a recurrent encoder to progressively encode image compression. In recent years, convolutional autoencoders have been studied extensively, including non-linear transformations (e.g., generalized division normalization) [7], differentiable quantization (e.g., soft-to-hard quantization, and uniform noise approximation), hyper-prior [10] probability models to estimate entropy in end-to-end DNN image compression frameworks. Recently, the context-adaptive [16] and coarse to fine hyper-prior [10] entropy models have been designed to further improve the distortion rate performance and exceed the traditional image codec. Rate-Distortion Optimization [17] is applied to minimize the Lagrangian cost $R + \lambda D$ in end-to-end training. Here, R is the entropy rate, and D is the distortion measured by mean square error (MSE) or multi-scale structural similarity (MS-SSIM).

2.2 Learned Video Compression

Deep learning is also getting more and more attention in video compression. To improve the coding efficiency of manual standards (e.g., H.264 and H.265), many methods have been proposed [6], [18] to replace components in H.264 with DNN. Among them, [19] used DNN in motion compensation score interpolation, [20] proposed a DNN model for frame predictions. In addition, [21], [22] use DNN to enhance the H.265 loop filter. However, these methods can only improve the performance of a specific component of the video compression framework, but they fail to optimize the video frames together. Inspired by the success of learning image compressions, some learning-based video compression methods are proposed in [2], [25]. However, [20] and [1] still use some manual strategies, such as block matching for motion estimation and compensation, so the entire compressed frame cannot be optimized in an overall manner. Recently, several end-to-end DNN frames have been proposed for video compression [24], [25]. Specifically, [8] proposed a deep video compression (DVC) method, which uses optical flow for the motion estimation and uses two auto-encoders to compress motion and residual separately. DVC outperformed on H.264 mainly at high bit rates, but the coding efficiency dropped unexpectedly at low bit rates as reported. Our model proposes a Spatial-temporal compression method for videos, and it outperforms from low to high bitrates as compared to H.264.

2.3 Attention mechanism

In general, attention can be thought of as a guideline for allocating available processing resources to the most informative aspects. To rescale the feature maps, it's frequently paired with a gating function (e.g., sigmoid). With a trunk-and-mask attention mechanism, [26] presented a residual attention network for image classification. The attention module is also used by [27] in video compression to improve performance. Overall, the goal of these efforts is to direct the network's attention to the regions of interest. However, there has been an investigation into using the attention mechanism to learn the spatial correlation of different sampling densities to increase video compression efficiency.

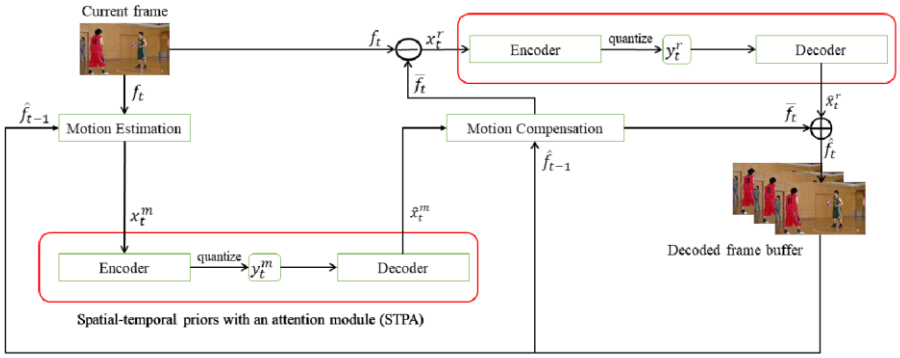

Fig. 1. The framework of our STVC approach. The modules with red colour are proposed STPA network shown in Fig. 2. We use hyperprior, which is illustrated in Fig. 2, is applied on the latent representations y_t^m and y_t^r to reduce the spatial redundancy.

3 THE PROPOSED STVC APPROACH

3.1 Framework

Fig. 1 shows the main components of the framework and the relationships among them. The framework is inspired by traditional video codecs H.264 and H.265. A video can be thought of as a collection of frames f_1, f_2, ...f_{t-1}, f_t. So, we define the current video sequences and compressed frames as $\{f_t\}_{t=1}^{T}$ and $\{\hat{f}_t\}_{t=1}^{T}$, respectively. To reduce the redundancy in frames, motion estimation is required. We apply the optical flow network [28] to estimate the temporal motion between the current frame and the previously compressed frame. In our framework, we use the same motion compensation method as [8]. In the following, the residual x_t^r, between f_t and the motion compensated frame $\bar{f}_t$ can be obtained and compressed by another STPA. Using the compressed residual as $\hat{x}_t^r$, we can reconstruct the compressed frame $\hat{f}_t = \bar{f}_t + \hat{x}_t^r$.

3.2 Spatial-temporal priors with an attention module (STPA)

In STVC, we apply two STPAs to compress x_t^m and x_t^r for motion and residual compression, respectively. Since the two STPAs share the same architecture, we denote both x_t^m and x_t^r by x_t in this section for simplicity. The compression process in STPA can be formulated by

$$y_t = E(x_t; \phi_E)$$

$$\hat{y}_t = Q(y_t)$$

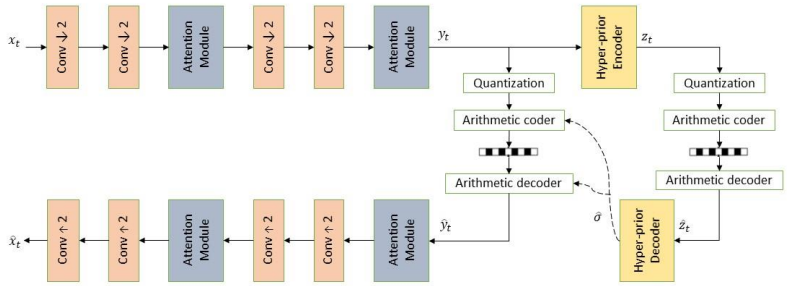

Fig. 2. The architecture of the proposed STPA network. In convolutions layers, ↑2 and ↓2 indicate up and down sampling with the stride of 2, respectively. In STPA, the filter sizes of all convolutional layers are set as 3 × 3 when compressing motion and set as 5 × 5 for residual compression. The filter number of each layer is set as 128. Attention modules are the same as in [29].

$$\hat{x}_t = D(\hat{y}_t; \theta_D) \tag{1}$$

Here x_t, $\hat{x}_t$, y_t, $\hat{y}_t$ denote the current input frame, reconstructed frame, the latent variables, and the quantized latent variables, respectively. Notation E and D denote the encoder and decoder, respectively, and ϕ and θ correspond to their parameters. Notation Q denotes real round-based quantization in the inference stage.

In the training process, considering that non-differentiable quantization will result in the inability to backpropagate the gradient, the work uses a uniform noise to replace the quantization here. When compressing the t-th frame, the auto-encoders map the input x_t to a latent representation.

$$\tilde{y}_t = U(y_t)$$

$$\tilde{x}_t = D(\tilde{y}_t; \theta_D) \tag{2}$$

Where $\tilde{y}_t$ and $\tilde{x}_t$ represent the latent variables with uniform noise added and its decoding reconstruction. Notation U denotes adding uniform noise in the training stage.

Taking the inputs of only the current x_t and y_t to the encoder and decoder, they fail to take advantage of the spatial redundancy and are not able to recover missed details in the latent variables y.

On the contrary, the proposed STPA includes hyperprior and attention in both the encoder and decoder. The architecture of the STPA network is illustrated in Fig. 2. We follow [10] to use four 2×down-sampling convolutional layers with the activation function of GDN [7] in the encoder of STPA. In the middle of the four convolutional layers and bottleneck of quantized features, we insert an attention module [29] to reduce the artifacts and recover missed details. Therefore, the model can pay more attention to complex regions to improve coding performance. Therefore, the proposed STPA

generates latent representation based on the current frame as well the as previous frame. Similarly, the STPA decoder also has an attention module with the four 2×up-sampling convolutional layers with IGDN, and thus also reconstructs $\hat{x}_t$ from both the current and previous latent representations.

To reduce the spatial redundancy in the latent variables y_t, an auxiliary hyperprior network [10] encodes its structural information z_t. Formulated by

$$z_t = E_h(y_t; \phi_h)$$

$$\hat{z}_t = Q(\mathrm{z_t})$$

$$p_{\hat{y}_t|\hat{z}_t}(\hat{\mathrm{y}}_\mathrm{t}|\hat{z}_t) \leftarrow \mathrm{D_h}(\hat{z}_t;\ \theta_\mathrm{h}) \tag{3}$$

Where E_h and D_h denote the encoder and decoder of this hyperprior network, and ϕ_h and θ_h correspond to their trainable parameters. $p_{\hat{y}_t|\hat{z}_t}(\hat{y}_t|\hat{z}_t)$ are estimated distributions conditioned on z_t. There is no prior for z_t, so a factorized density model ψ is used to encode z_t as

$$p_{\hat{z}_{it}|\psi}(\hat{z}_{it}|\psi) = \Pi_i \left(p_{z_{it}|\psi}(\psi) * U\left(-\frac{1}{2}, \frac{1}{2}\right)\right)(\hat{z}_{it}) \tag{4}$$

Where z_{it} denotes the i-th element of z at time t, and i specifies the position of each element or each signal.

3.3 Loss Function

The purpose of our video compression framework is to reduce distortion between the original input frame, f_t and the compressed frame $\hat{f}_t$ while reducing the number of bits utilized for encoding the video. As a result, we propose the rate-distortion optimization problem below.

$$\mathrm{L} = \lambda \mathrm{D} + \mathrm{R} = \lambda \mathrm{d}(f_t, \hat{f}_t) + \hat{y}_t^m + \hat{y}_t^r \tag{5}$$

We employ mean square error (MSE) in our approach, and $\mathrm{d}(f_t, \hat{f}_t)$ specifies the distortion between f_t and $\hat{f}_t$. The amount of bits utilized to encode the representations is represented by I(). Both the residual representation $\hat{y}_t^r$ and the motion representation $\hat{y}_t^m$ should be encoded into the bitstreams in our technique. λ is the Lagrange multiplier that determines the trade-off between the number of bits and distortion.

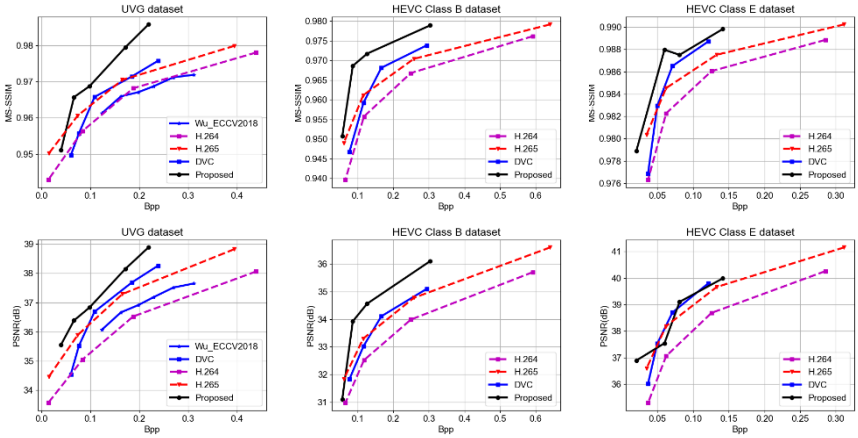

Fig. 3. Rate-distortion performance of the proposed approach compared with the learning-based video codec in [30], DVC [8], and traditional video compression method H.264 [4] and H.265 [5] approaches on the UVG and JCT-VC datasets.

4 EXPERIMENTS

4.1 Training the proposed network

Our proposed approach is trained using the Vimeo-90k dataset [31]. The dataset is built for video denoising, deblocking, and super-resolution. Each training sample video has seven frames. The first frame is compressed as an I-frame, while the remaining six frames are compressed as P-frames. To evaluate compression quality, we use the Multiscale Structural Similarity (MS-SSIM) [32] index and the Peak Signal-to-Noise Ratio (PSNR), and then train the model with settings that are optimized for MS-SSIM and PSNR, respectively. To train the PSNR model, it uses the Mean Square Error (MSE). We set $\lambda = 1024$ for MS-SSIM and PSNR. The Adam optimizer [33] is utilized for training. The initial learning rate is set as 10^{-4} for loss function. We use Bpp (bits per pixel) to express the required bits for each pixel in the current frame to measure the number of bits for encoding the representations.

4.2 Evaluating the performance

The evaluations are being carried out to ensure that our proposed model is effective. The performance is evaluated using the JCT-VC [4] (Classes B and E) and UVG [34] datasets. JCT-VC Classes B and UVG have 1920 x 1080 resolutions. The UVG dataset has a GOP (group of pictures) size of 12, and the HEVC dataset has a GOP size of 10, respectively. We compare our method to that of DVC [8], [30], H.264 [4], and

Table 1. BDBR performances with Proposed, DVC, and H.265 model when compared with H.264.

Dataset	MS-SSIM			PSNR		
	Proposed	DVC	H.265	Proposed	DVC	H.265
UVG(Average)	**-38.62**	-16.46	-26.19	**-39.24**	-37.34	-36.19
Class B(Average)	**-29.59**	-29.09	-28.31	**-33.17**	-27.92	-31.73
Class E(Average)	**-35.51**	-33.16	-29.52	**-36.01**	-22.23	-35.54
Average	**-34.57**	-26.24	-28.17	**-36.14**	-29.26	-34.30

H.265 [5]. We use FFmpeg in very fast mode and the settings in [30] to make compressed frames using H.264 and H.265.

Rate-distortion curve- The rate-distortion curves for the JCT-VC and UVG datasets are shown in Fig. 3. The bit rate is determined using Bpp, while the quality is measured using MS-SSIM and PSNR. Overall, the proposed MS-SSIM model outperforms DVC [8], [30], H.264, and H.265, as illustrated in Fig. 3. DVC is comparable with H.265 at low bit rates on the JCT-VC dataset but our approach's rate-distortion curves clearly outperform DVC [8], from low to high bitrates on the JCT-VC dataset. Also, it can be demonstrated that our PSNR model outperforms DVC [8], [30], H.264, and H.265 on all videos (average).

Bit-rate difference- In addition, we evaluated Bjøntegaard Delta Bitrate (BDBR) [35] using H.264 anchors. BDBR calculates the average bit rate difference compared to the anchor point, and a lower BDBR value indicates better performance. Table 1 shows the BDBR calculated using MS-SSIM and PSNR, where a negative number indicates that the bit rate is lower than that of the anchor, which means that it is better than H.264, and the bold number is the best result among all learning methods.

In Table 1, in order to fairly compare MS-SSIM with optimized methods DVC [8], H.264, and H.265, we first report the BDBR of our model based on MS-SSIM. In UVG and JCT-VC types B and E, our model is even significantly better than the DVC method in terms of MS-SSIM. As shown in Table 1, our model is better than H.264 in MS-SSIM, with an average BDBR of 34.57 %, which is also better than DVC (BDBR = 26.24 %). In terms of PSNR, Table 1 shows that our PSNR model outperforms H.264 (very fast LDP) from low to high bit rates on the UVG and JCT-VC test sets. The BDBR results calculated by PSNR in Table 1 also show that the bitrate achieved by our method is 36.14% lower than H.264 (very fast LDP). Please note that as far we know, there are no learned video compression methods that have exceeded the default setting of H.265 in PSNR. Our proposed method can be further developed to improve the performance of the next generation learning video compression and to help gradually catch up with manual standards.

5 CONCLUSION AND FUTURE WORK

This paper has proposed a spatial-temporal video compression network (STVC). Specifically, we proposed an auto-encoders style network with an attention module to compress motion and residual, fully exploring the spatial and temporal correlation in video frames. We proposed a rate-distortion optimization framework to train a single Spatial-temporal autoencoder for reconstruction loss. Key novelty laid on the accurate motion representation for exploiting temporal correlation and hyper-priors is leveraged to improve the spatial correlation and entropy coding efficiency.

We evaluated our methods and reported the performances among the traditional H.264/AVC, H.265/HEVC, and learning-based DVC and [30]. Our approach offered consistent gains over existing methods across a variety of contents and bit rates, but the PSNR model on the default setting of H.265 did not beat the performance, so it needs further research and development. Moreover, the proposed approach achieved significant performance at the cost of higher encoding complexity. Another possible focus for future work is to study reducing complexity and the trade-off between complexity and rate-distortion performance. For example, the proposed network may be sped up by reducing the number of layers and channels in the autoencoders and the motion compensation network or by utilizing a more time-efficient optical flow network for motion prediction [1].

References

[1] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, “DeepCoder: A Deep Neural Network Based Video Compression.”

[2] Z. Chen, T. He, X. Jin, and F. Wu, “Learning for Video Compression,” Apr. 2018, doi: 10.1109/TCSVT.2019.2892608.

[3] S. Aramvith and M.-T. Sun, “MPEG-1 AND MPEG-2 Video Standards,” 1999.

[4] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003, doi: 10.1109/TCSVT.2003.815165.

[5] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012, doi: 10.1109/TCSVT.2012.2221191.

[6] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, “Reducing Complexity of HEVC: A Deep Learning Approach,” Sep. 2017, doi: 10.1109/TIP.2018.2847035.

[7] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end Optimized Image Compression,” Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.01704

[8] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “DVC: An End-toend Deep Video Compression Framework,” Nov. 2018, [Online]. Available: http://arxiv.org/abs/1812.00101

[9] J. Pessoa, H. Aidos, P. Tomas, and M. A. T. Figueiredo, “End-to-End Learning of Video Compression using Spatio-Temporal Autoencoders,” in *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, Oct. 2020, vol. 2020-October. doi: 10.1109/SiPS50750.2020.9195249.

[10] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” Jan. 2018, [Online]. Available: http://arxiv.org/abs/1802.01436

[11] Y. Hu, W. Yang, and J. Liu, “Coarse-to-Fine Hyper-Prior Modeling for Learned Image Compression.” [Online]. Available: https://huzi96.github.io/coarse-to-fine-compression.html.

[12] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “Video Summarization Using Deep Neural Networks: A Survey,” Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.06072

[13] Y. Xu *et al.*, “GIF Thumbnails: Attract More Clicks to Your Videos,” 2021. [Online]. Available: www.aaai.org

[14] N. Zou *et al.*, “End-to-End Learning for Video Frame Compression with SelfAttention,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.09226

[15] G. Toderici *et al.*, “Full Resolution Image Compression with Recurrent Neural Networks,” Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.05148

[16] J. Lee, S. Cho, and S.-K. Beack, “CONTEXT-ADAPTIVE ENTROPY MODEL FOR END-TO-END OPTIMIZED IMAGE COMPRESSION.” [Online]. Available: https://github.com/JooyoungLeeETRI/CA_Entropy_Model.

[17] “Sullivan - RD Opt for Video”.

[18] T. Li, M. Xu, C. Zhu, R. Yang, Z. Wang, and Z. Guan, “A Deep Learning Approach for Multi-Frame In-Loop Filter of HEVC,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 28, no. 11, pp. 5663–5678, Nov. 2019, doi: 10.1109/TIP.2019.2921877.

[19] J. Liu, S. Xia, W. Yang, M. Li, and D. Liu, “One-for-All: Grouped Variation Network-Based Fractional Interpolation in Video Coding,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2140–2151, May 2019, doi: 10.1109/TIP.2018.2882923.

[20] H. Choi and I. v. Bajic, “Deep Frame Prediction for Video Coding,” Dec. 2018, [Online]. Available: http://arxiv.org/abs/1901.00062

[21] Y. Dai, D. Liu, and F. Wu, “A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding,” Aug. 2016, doi: 10.1007/978-3-31951811-4_3.

[22] T. Li, M. Xu, R. Yang, and X. Tao, “A DenseNet Based Approach for Multiframe In-loop Filter in HEVC,” in *Data Compression Conference Proceedings*, May 2019, vol. 2019-March, pp. 270–279. doi: 10.1109/DCC.2019.00035.

[23] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, “DeepCoder: A Deep Neural Network Based Video Compression.”

[24] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learning Image and Video Compression through Spatial-Temporal Energy Compaction.”

[25] R. Yang, F. Mentzer, L. van Gool, and R. Timofte, “Learning for Video Compression with Hierarchical Quality and Recurrent Enhancement,” Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.01966

[26] F. Wang *et al.*, “Residual Attention Network for Image Classification,” Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.06904

[27] M. Zhao, Y. Xu, and S. Zhou, “Recursive Fusion and Deformable Spatiotemporal Attention for Video Compression Artifact Reduction,” in *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, Oct. 2021, pp. 5646–5654. doi: 10.1145/3474085.3475710.

[28] A. Ranjan and M. J. Black, “Optical Flow Estimation using a Spatial Pyramid Network,” Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.00850

[29] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules,” Jan. 2020, [Online]. Available: http://arxiv.org/abs/2001.01568

[30] C.-Y. Wu, N. Singhal, and P. Krähenbühl, “Video Compression through Image Interpolation,” Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.06919

[31] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video Enhancement with Task-Oriented Flow,” Nov. 2017, doi: 10.1007/s11263-018-01144-2.

[32] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “MULTI-SCALE STRUCTURAL SIMILARITY FOR IMAGE QUALITY ASSESSMENT.”

[33] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[34] A. Mercat, M. Viitanen, and J. Vanne, “UVG dataset: 50/120fps 4K sequences for video codec analysis and development,” in *MMSys 2020 - Proceedings of the 2020 Multimedia Systems Conference*, May 2020, pp. 297–302. doi: 10.1145/3339825.3394937.

[35] P. Hanhart and T. Ebrahimi, “Calculation of average coding efficiency based on subjective quality scores.” [Online]. Available: http://mmspg.epfl.ch/scenic