2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506275

DEEP VIDEO COMPRESSION FOR INTERFRAME CODING

David Alexandre[1], *Hsueh-Ming Hang*[1], *Wen-Hsiao Peng*[1], *Marek Domański*[2]

[1]National Yang Ming Chiao Tung University, Taiwan [2]Poznań University of Technology, Poland

ABSTRACT

A typical learning-based video compression scheme consists of motion coding and residual coding. In this paper, our deep video compression features a motion predictor and refinement networks for interframe coding. To save the bits for transmitting motion information, our scheme performs local motion prediction and sends only the differential motion vectors to the decoder. In the residual coding, we couple the residual decoder with the refine-net to reduce residual signal bits. The experiments show that our work can produce a very competitive coding performance compared to the other learning-based predictive video codecs.

Index Terms— *Deep learning, video compression, predictive coding, video extrapolation*

1. INTRODUCTION

The motion-compensated transform coding architecture has been adopted by the modern video coding standards such as H.264 [1] and H.265 [2]. Similar concepts are carried to the development of learning-based video coding schemes. The encoder transmits the motion information (motion vectors) and the residual information to the decoder. Both encoder and decoder use the motion information to warp the previous frame to form the prediction of the current frame. This is called *predictive coding* or P-frame coding.

Influenced by the conventional coding structures, a typical learning-based video coding scheme uses the Deep Neural Network (DNN) to perform motion information coding and image residual coding. To reduce reconstructed image artifacts, the so-called refinement networks (refine-net) have been developed to repair the signals coming out of the decompressors.

Targeting at predictive coding, we propose a learning-based video compression scheme in this paper. Part of its elements come from our previous work [3]. The key contributions of this work are as follows.

1. This is a complete interframe video coding scheme that employs I- and P-frames within GOPs (Groups of Pictures), whereas [3] is a single P-frame coding scheme based on the original reference frame.
2. We propose a motion vector extrapolation net, which enables our scheme to transmit only the differential motion vectors (or optical flow) to the decoder.
3. We continue using the residual compressor and refine-net pair as in [3]. A multi-phase training process is carefully crafted to produce an efficient system.

Our system incorporates the motion extrapolation concept into a neural-net based video codec. With fine-tune the refine-nets for motion residuals and image residuals, our system can provide a coding performance competitive to the state-of-the-art solutions.

This paper is organized as follows. The related works are covered in Section 2. The details of our proposed method are described in Section 3. Next, we present the experiments and evaluation in Section 4 and a brief conclusion in Section 5.

2. RELATED WORKS

Most learning-based video compression schemes adopt the motion-compensated residual coding structure. For example, Lu, *et al.* [4] proposed the DVC scheme, which uses Spy-Net for motion estimation and use the hyperprior autoencoder [5] for residual coding. Hu, *et al.* [6] uses the previous motion information with a resolution adaptive flow coding for motion estimation and coding. They use Minnen, *et al.* [7] autoregressive hyperprior model for residual coding. Lin, *et al.* [8] proposed the M-LVC scheme with motion field refine-net before the MC-net (motion compensation). For the residual coding, they use the hyperprior autoencoder designed by Balle, *et al.* [5].

One powerful DNN tool commonly used in image coding is the inclusion of refine-net; it helps to enhance the quality of decompressed image. The network architectures commonly used for doing this job is *residual block* and *attention block*. For example, Zamir, *et al.* [9] constructed a multi-scale architecture with residual block and attention mechanism for image enhancement. Vu, *et al.* [10] proposed an image enhancement network with residual blocks for the low-resolution input images.

Inspired by the RaFC [6] and M-LVC [8] schemes, we perform local motion vector extrapolation and transmit only the differential motion vectors. In addition, in contrast to the previous approaches whose refine-nets act as a post-processor, our refine-net is designed to be a partner of the compressor network to reduce the transmission bitrate when producing good reconstruction quality.

3. PROPOSED METHOD

3.1. Proposed Architecture

Fig. 1 shows our deep video compression architecture. Compared to the other learning-based video codecs, our contributions are the motion predictor-net and the decoded image refine-net. These components implement interframe video coding (IPPP scheme) by cooperation with motion extractor-net (multi-scale motion estimator based on PWC-

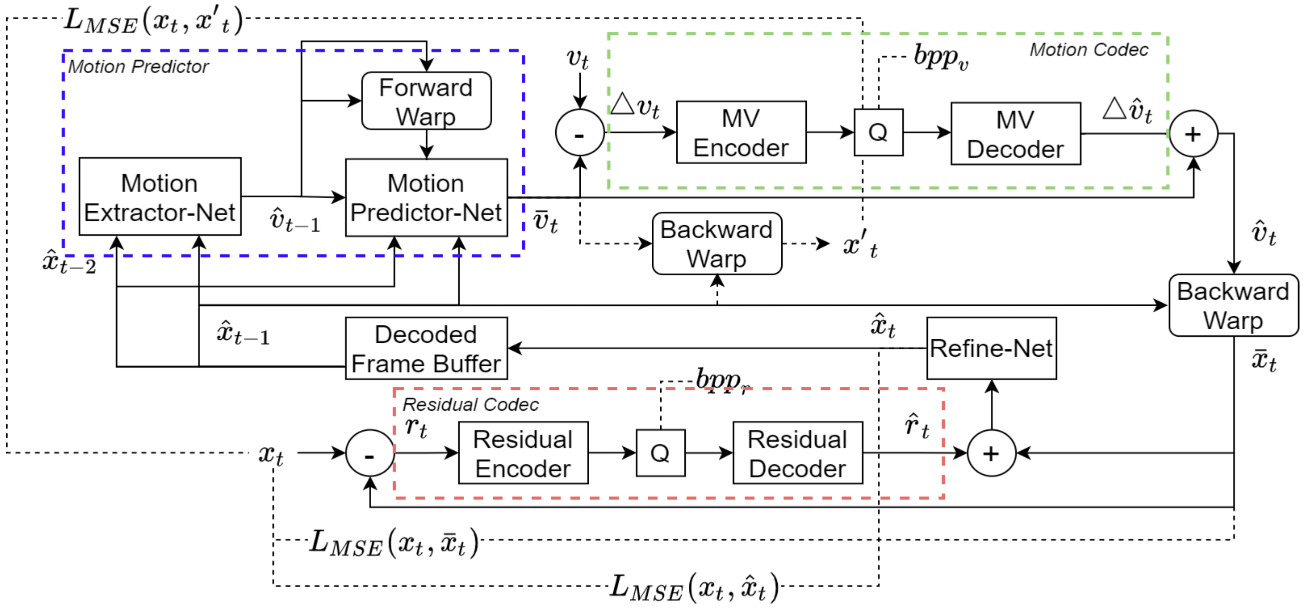

Figure 1. Our system consists of 3 subsystems: *motion (optical flow) predictor*, *motion (differential motion vector) codec*, and *residual (motion-compensated frame difference) codec*. The motion extractor-net estimates the optical flow, $\hat{v}_{t-1}$, based on frames $\hat{x}_{t-1}$ and $\hat{x}_{t-2}$. Then, we predict the optical flow for frame *t*, $\bar{v}_t$, using forward warping and polish it using a *hierarchical motion refinement-net*. The differential motion vectors $\triangle v_t = v_t - \bar{v}_t$ are compressed and send to the decoder. The identical motion predictor at the decoder reproduces $\bar{v}_t$, and then the encoded optical flow, $\hat{v}_t$ is recovered. The motion-compensated image residual, r_t, is compressed and transmitted. At the end, the decoder reconstructs the target frame, $\hat{x}_t$, using the refine-net. The variables (x'_t, bpp_v, bpp_r) connected by dotted lines are calculated and used only in the training process.

Net) and the residual compressors. For intraframe coding, we adopted the learning-based single image compressor proposed by Cheng, *et al.* [11].

For the interframe coding, we divide our architecture into two main parts, *motion information coding* and *image residual coding*. The motion coding subsystem is responsible to produce motion-compensated frames. Using two previously reconstructed frames retrieved from the decoded frame buffer, a *motion extractor* network produces the *backward* optical flow between two input frames. Its structure was described in [3]. The prediction of motion vectors (optical flow) is a well-known concept in the conventional coding systems. To reduce the number of transmitted bits, we designed a *motion predictor-net* to perform motion vector (optical flow) extrapolation, which estimates the motion information needed to produce the target frame. Then, we only need to transmit the *differential optical flow*, For the residual coding, the image residual signals between the target frame and the motion-compensated frame is compressed using the residual compressor.

3.2. Motion Information Coding

As shown in Fig. 1, our motion-coding mechanism consists of *motion extractor network*, *motion predictor network*, and *motion compressor*.

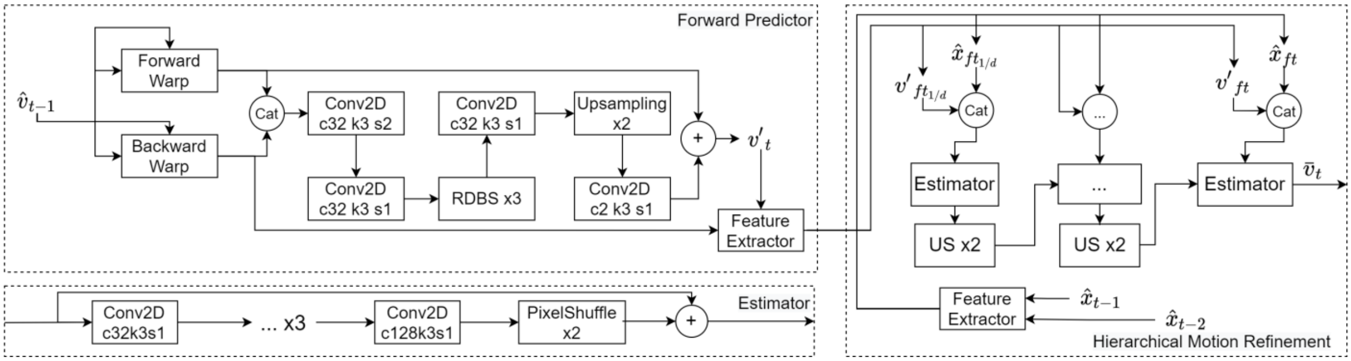

Figure 2. Motion predictor-net consists of forward predictor and hierarchical motion refinement. It includes CNN layers and residual dense blocks (RDBS) to produce a predicted optical flow $\boldsymbol{v}$ ´$_{\boldsymbol{t}}$. The motion vector forward warping creates holes in the predicted optical flow. Hence, the backward warping provides the background optical flow to fill up holes. Using multiple scales ($1/d$) of predicted optical flow in feature domain, our hierarchical refine-net progressively enhances the final form of predicted optical flow $\bar{\boldsymbol{v}}_{\boldsymbol{t}}$.

Motion Extractor. Given the input reference frames $\hat{x}_{t-2}$, $\hat{x}_{t-1}$ and the target frame x_t, the motion extractor network produces the backward optical flows $\hat{v}_{t-1}$ and v_t from the $\hat{x}_{t-2}$, $\hat{x}_{t-1}$ pair and the $\hat{x}_{t-1}$, x_t pair, respectively. This motion extractor is essentially the hierarchical motion field generator proposed in our previous work [3]. Firstly, the initial optical flow is generated by the PWC-net from Sun, *et al.* [12] but, similar to DVC, it is then processed by a series of multi-scale refinement networks to improve the motion accuracy. Then, the optical flow $\hat{v}_{t-1}$ is fed to the motion predictor-net to produce an estimate of v_t, $\bar{v}_t$. Then, the difference between v_t and $\bar{v}_t$, $\triangle v_t$, is compressed and transmitted to the receiver.

Motion Predictor. Inspired by motion vector prediction in conventional video coding, we designed a local motion extrapolator that uses two previous decoded frames. Our motion predictor subsystem takes only 3 inputs: $\hat{x}_{t-1}$, $\hat{x}_{t-2}$ and $\hat{v}_{t-1}$. There are two components inside our motion predictor subsystem: forward motion predictor, and multi-scale motion vector refinement-net as shown in Fig. 2. We reverse the directions of $\hat{v}_{t-1}$ to forward-warp itself to obtain a rough estimate of the target optical flow, v_t'. Then, we use a multi-scale feature extractor to convert the reference frames $\hat{x}_{t-1}$, $\hat{x}_{t-2}$, and v_t' and the backward-warped optical flow to feature maps at different scales. These feature maps are fed to a hierarchical motion refine-net to produce the final predicted optical flow $\bar{v}_t$. In training the motion predictor sub-system, we use the *MSE* between the motion compensated frame using our predicted flow x_t' (Fig.1) and the target frame x_t.

Motion Compressor. We use the compressor structure in our previous work [3], which was modified from the network architecture of Minnen, *et al.* [7]. The channel number for encoder, decoder and the compressed feature maps is all set to 128. For the inputs, we normalize the optical flow to a range of (-1, 1). To recover it at the decoder side, we transmit the normalization scaling factor to the decoder. The motion coding subsystem is trained using the rate-distortion loss, $L_v = \lambda * D_v + R_v$, where $D_v = MSE(\bar{x}_t, x_t)$ is the distortion and $R_v = bpp_v$ is the bitrate estimated by the hyperprior autoencoder. In the training step, we use the additive uniform noise technique to replace the quantization process to avoid the vanishing gradient problem. In inferencing, our entropy coding is the Range Asymmetric Numeral System (rANS) from Duda, *et al.* [14]. The same system is also used in encoding the image residual signals.

3.3. Image Residual Coding

The image residual coding consists of residual compressor and refine-net. Our residual compressor encodes the motion-compensated frame difference (MCFD) or image residuals between the motion-compensated frame and the target frame.

Residual Compressor. Given a residual *r* between the target image x_t and motion-compensated frame $\bar{x}_t$, we perform residual coding to produce the decoded residual $\hat{r}_t$ by using the network architecture in our prior work [3]. In this setup, the residual compressor produces 128 channels of feature maps for quantization and entropy coding. Our residual compressor accepts input values in the range of (-1,1). We first pretrained the compressor network before using it in the end-to-end training phases with rate-distortion loss, $L_r = \lambda * D_r + R_r$. The D_r term is calculated using $MSE(\hat{x}_t, x_t)$ or $MSSSIM(\hat{x}_t, x_t)$, and R_r is estimated by the hyperprior network during the training, bpp_r.

Refine-net. To further improve reconstructed video quality and coding efficiency, we design an image refinement network integrated with the residual compressor. We

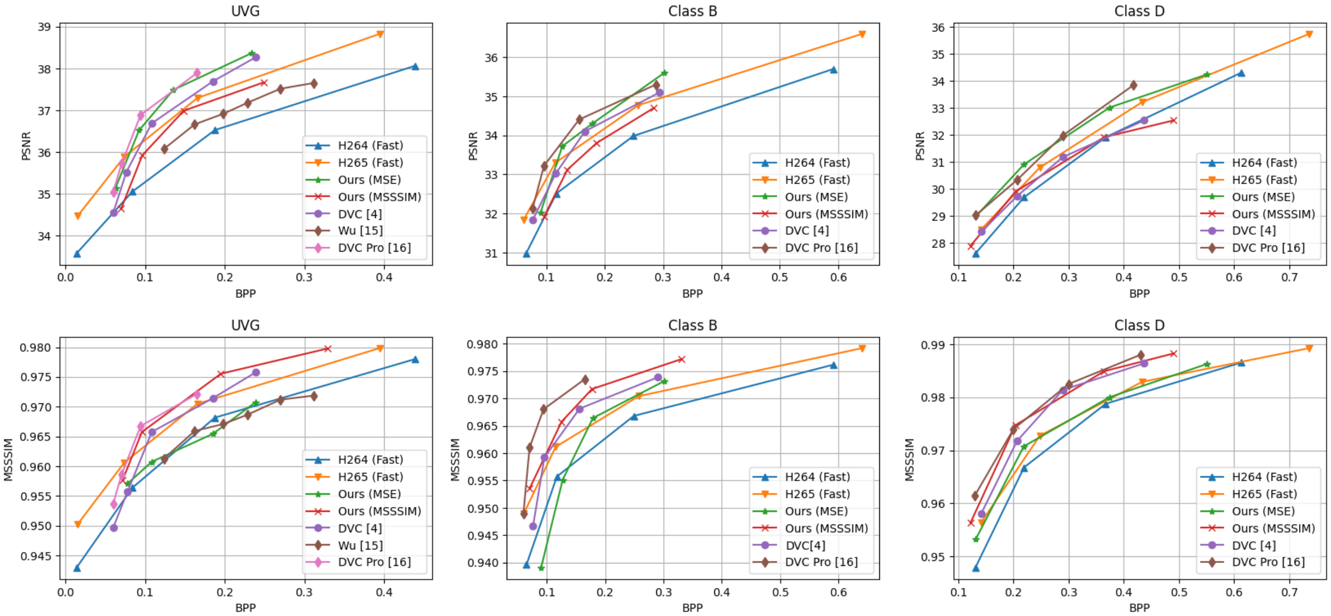

Figure 3. We evaluate the results on HEVC Class B, class D, and UVG datasets in terms of MSE and MS-SSIM.

continue to use the refine-net structure developed in our prior work [3].

3.4. Training Phases

We split the training process into three phases. To speed up the process, we generate optical flows and residual signals in advance using the training set described in Section 4.1. In preparation, we train the motion extractor-net (including the hierarchical motion field generator) first. Then, in the first phase, we train motion predictor-net using D_E. In the second phase, the motion codec is trained using $\lambda_v * D_v + R_v$. Also, the residual codec is trained separately using $\lambda_r * D_r + R_r$.

$$D_E = MSE(x'_t, x_t),\ D_v = MSE(\bar{x}_t, x_t)$$
$$D_r = MSE(\hat{x}_t, x_t) \ / \ MSSSIM(\hat{x}_t, x_t)$$
$$L = \lambda_v * D_v + \lambda_r * D_r + R_v + R_r \qquad (1)$$

In the final training phase, we train the entire system together but the motion predictor subsystem is detached from the entire system in the sense that its parameters are adjusted to minimize D_E. And the rest of the system is adjusted to minimize the joint loss function defined by (1). We found that the regularization term D_v helps in improving the training process convergence and the overall coding performance. Empirically, we keep a ratio between two λ parameters at different bit rates, $\lambda_v = 0.2 * \lambda_r$.

4. EXPERIMENTS

Our system is evaluated using the popular datasets for video coding benchmarks, such as HEVC [2] and UVG [17] dataset. In addition, we conducted analysis and example for motion predictor on the HEVC class D datasets.

4.1. Dataset and Training Setup

Our training and validation were performed using 91,701 7-frame sequences from Vimeo septuplet dataset [14]. The experiments were done using the DPP mode on 4 x NVIDIA Tesla V100. The evaluation was performed for the interframe IPPP scenario on the HEVC classes B and D, and UVG datasets. To have a fair comparison, we follow the evaluation setting in DVC [4]. For the HEVC dataset, we use GOP=10. For the UVG dataset, GOP=12. In both cases, we run the interframe coding for the first one hundred frames.

4.2. Evaluation Results

Fig. 3 shows the evaluation results of our scheme. We compare our results with the conventional codecs, H.264, H.265 and the IPPP-based learning-based video coding schemes, DVC. The RD curves for H.264 and H.265 are obtained from the DVC paper [4] (in *very fast* setting). Two systems, our-MSE and ours-MSSSIM, were trained using the MSE and MS-SSIM distortion metrics, individually, in the final end-to-end training phase. Our system offers a slightly higher performance compared to DVC.

Network Complexity and Evaluation Time. Our video coding uses about 19M network parameters. For class D (416x240 pixels), our system takes approximately 0.673s per frame for encoding and 0.21s for decoding; for class B and UVG (1920x1080 pixels) video test sequences, it takes approximately 1.423s for encoding and 0.552s for decoding per frame.

5. CONCLUSION

A learning-based video compression system is presented for interframe coding. We propose a motion predictor-net in this system, which predicts the motion vectors for the target frame. It reduces the transmitted motion information by sending the differential motion vectors. We also designed a refine-net working together with the residual codec. Based on the evaluation results, we conclude that, with respect to the coding performance, our method is very competitive as compared to other learning-based video codecs.

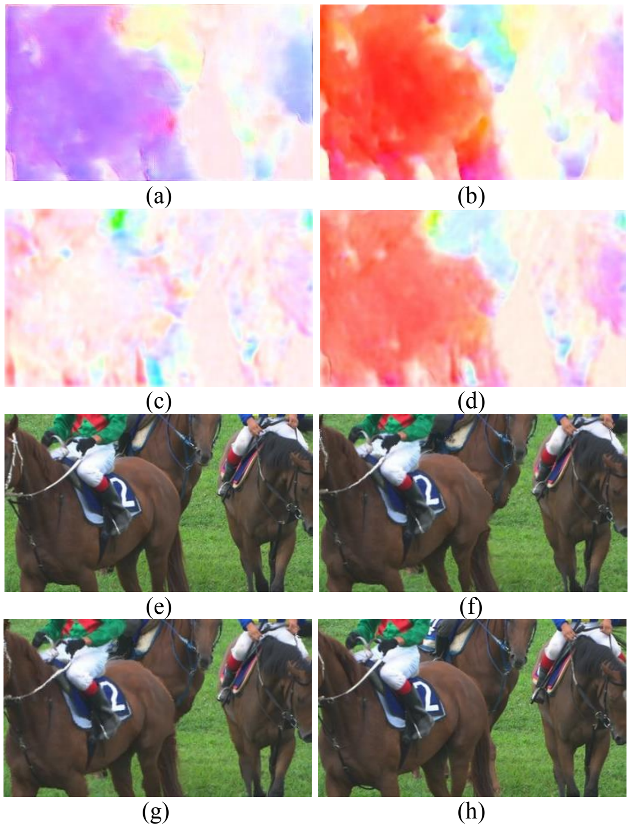

Figure 4. (a) Estimated optical flow, $\hat{v}_{t-1}$, (b) Predicted optical flow, $\bar{v}_t$, (c) Transmitted differential optical flow, (d) Reconstructed optical flow, $\hat{v}_t$, (e) Reference frame, $\hat{x}_{t-1}$, (f) Warped frame, $warp(\hat{x}_{t-1}, \bar{v}_t)$, (g) Motion-compensated prediction $\bar{x}_t$ from $warp(\hat{x}_{t-1}, \hat{v}_t)$, (h) Target frame x_t.

6. ACKNOWLEDGEMENT

This work is partially supported by the Ministry of Science and Technology, Taiwan under Grant MOST 108-2923-E-009-004 and 109-2634-F-009-020, and National Centre for Research and Development (NCBiR) of Poland under Project ECIVAD. We also like to thank the support of Pervasive AI Research (PAIR) Labs, National Yang Ming Chiao Tung University and National Center for High-performance Computing, Taiwan for providing computational facilities.

7. REFERENCES

[1] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H. 264/AVC video coding standard,” *IEEE Transactions on circuits and systems for video technology*, 13(7), pp.560-576, 2003.

[2] 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*, 22(12), pp.1649-1668, 2012.

[3] D. Alexandre, and H-M. Hang, “Learned video codec with enriched reconstruction for CLIC P-frame coding,” arXiv preprint arXiv:2012.07462. 2020.

[4] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “DVC: An end-to-end deep video compression framework,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11006-11015, 2019.

[5] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, . “Variational image compression with a scale hyperprior,” in *International Conference on Learning Representations*, 2018

[6] Z. Hu, Z. Chen, D. Xu, G. Lu, W. Ouyang, and S. Gu, “Improving deep video compression by resolution-adaptive flow coding,” in *European Conference on Computer Vision*, pp. 193-209, August 2020.

[7] D. Minnen, J. Ballé, and G.D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” *Advances in Neural Information Processing Systems*, pp. 10771-10780. 2018.

[8] J. Lin, D. Liu, H. Li, and F. Wu, “M-LVC: Multiple Frames Prediction for Learned Video Compression,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 3546-3554, 2020.

[9] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.H. Yang, and L. Shao, “Learning Enriched Features for Real Image Restoration and Enhancement,” arXiv preprint arXiv:2003.06792. 2020.

[10] T. Vu, C. V. Nguyen, T.X. Pham, T.M. Luu, and C.D. Yoo, “Fast and efficient image quality enhancement via desubpixel convolutional neural networks,” in *Proceedings of the European Conference on Computer Vision* (ECCV), 2018.

[11] 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*, pp. 7939-7948, 2020.

[12] 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*, pp. 8934-8943, 2018.

[13] J. Duda, “Asymmetric numeral systems: entropy coding combining speed of Huffman coding with compression rate of arithmetic coding,” arXiv preprint arXiv:1311.2540, 2013.

[14] T. Xue, B. Chen, J. Wu, D. Wei, and W.T. Freeman, “Video enhancement with task-oriented flow,” *International Journal of Computer Vision*, 127(8), pp.1106-1125, 2019.

[15] C.Y. Wu, N. Singhal, and P. Krahenbuhl, “Video compression through image interpolation,” in *Proceedings of the European Conference on Computer Vision*, 2018.

[16] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao and D. Xu, “An end-to-end learning framework for video compression,” in *Proceedings of the IEEE transactions on pattern analysis and machine intelligence*, 2020.

[17] A. Mercat, M. Viitanen, and J. Vanne, “UVG dataset: 50/120fps 4K sequences for video codec analysis and development,” in Proc. ACM Multimedia Syst. Conf., Istanbul, Turkey, June 2020.