Neural Network-based Error Concealment for B-Frames in VVC

Martin Benjak, Niklas Aust, Yasser Samayoa and Jörn Ostermann
Institut für Informationsverarbeitung
Leibniz Universität Hannover
Hannover, Germany
{benjak, austnik, samayoa, office}@tnt.uni-hannover.de

***Abstract*—In this paper we introduce an error concealment method for VVC that error-conceals B-frames based on the neural frame interpolation network RIFE. The network is trained using the BVI-DVC dataset to infer even full-HD frames. We integrate our proposed model in the VVC reference software VTM for its evaluation. The average error of a whole GOP with a single corrupted frame is decreased by 15% and 24% in terms of PSNR measurement compared to block matching and frame copy, respectively. To our knowledge, our approach is currently the best performing error concealment algorithm for single slice per B-frame settings.**

***Index Terms*—VVC, video coding, error concealment**

I. Introduction

The evolution of technologies for displaying and recording of video signals has been responding to the rising demand for higher resolution devices. To meet these demands, the state-of-the-art video coding standard Versatile Video Coding (VVC) [1] has been released in summer 2020. This is driving all types of communication systems to increase their capacity of conveying video. For instance, by 2022 nearly four-fifths of the world's mobile data traffic will be video [2]. Many applications like video surveillance, tele-medicine and smart car navigation systems require greater resolution video communication systems [3].

For the transmission and storage of video signals, the imperative systems are video coding, channel coding and communication systems. However, unidirectional video transmission imposes extra challenges because error-free output can not be guaranteed at the decoder side by any means. This forces the execution of error concealment (EC) algorithms in the video decoder to minimize the impact of errors that cannot be corrected by the channel decoder. It is worth noting that the impact of an uncorrected error increases with the coding efficiency. On the one hand, each video compression standard reaches a higher coding efficiency in comparison to its predecessors. On the other hand, the complexity of a suitable EC increases as well. Additionally, in the last two video coding standards, VVC and High Efficiency Video Coding (HEVC) [4], error resilience mechanisms have not been included and there is no suggestion for EC. These new standards assume error-free transmissions, which can not be guaranteed for real systems.

The problem of EC has been of great importance since the beginning of digital video communication systems. Several solutions have been proposed for standards prior to HEVC [5]–[8]. These algorithms were developed for codec settings using independently decodable macroblocks (MB), in which the spatio-temporal correlation of MBs is exploited to error-conceal lost MBs. HEVC and VVC abandoned the MB-based coding scheme, but similar behavior can implemented using independently decodable slices. Few EC algorithms for HEVC can be found in the literature [8]–[13]. These schemes address EC with analytical methods by exploiting spatio-temporal information available in the decoder to construct the lost portion of the video. Recently, neural network-based frame estimation [14], [15] yielded impressive results and can also be used for EC: Sankisa et al. trained a deep neuronal network to emulate EC for a single lost slice assuming a frame is divided in multiple slices [16]. Its performance was not measured within any coding standard. Benjak et al. implemented an EC algorithm based on the frame extrapolation network PredNet and integrated it into the reference software VVC Test Model (VTM) [17]. Since their approach is based on frame extrapolation, it can error-conceal P-frames and I-frames but not B-Frames.

In this paper we propose a machine learning-based EC algorithm for B-frames in VVC. We focus on applications which require high coding efficiencies for video communication systems over error prone channels. One slice per frame is assumed, such that just one erroneous bit in the encoded bit-stream can completely corrupt a whole video frame and produce the worst video quality degradation for dependent frames due to inter-prediction. Our model makes use of a neural frame interpolation network to generate an estimated version of any lost frame from already decoded frames within a group of pictures (GOP). The impact of the error-concealed frame on the video quality is evaluated within VTM. Currently, VTM has no capability to detect and error-conceal a lost slice, which means that our proposed EC algorithm is implemented and adapted to the VTM decoder. I-frames cannot be error-concealed using our proposed algorithm, but combined with the work of Benjak et al. [17], all types of frames can be error-concealed.

The remainder of this paper is organized as follows. In Section II, we present the proposed algorithm. In Section III, an evaluation and experimental results are given and Section IV provides a conclusion for this paper.

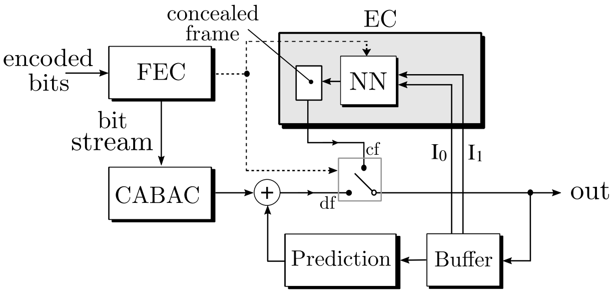


Fig. 1: Block diagram of VVC and NN method. FEC switches between concealed frame (cf) and decoded frame (df) depending on whether a frame is uncorrectably damaged.

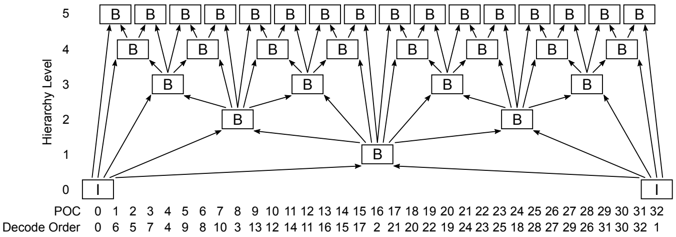

Hierarchy Level
5
4
3
2
1
0
B
I
POC 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
Decode Order 0 6 5 7 4 9 8 10 3 13 12 14 11 16 15 17 2 21 20 22 19 24 23 25 18 28 27 29 26 31 30 32 1

Fig. 2: GOP structure used for all experiments

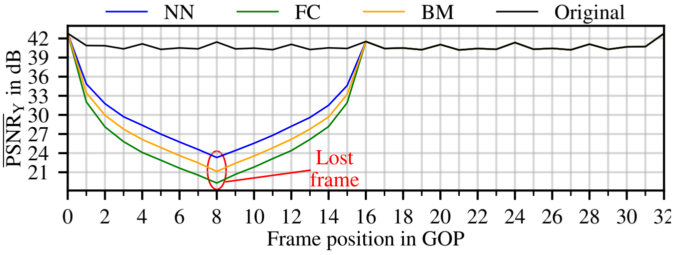


Fig. 3: $\overline{\mathrm{PSNR}}_{\mathrm{Y}}$ over all frames of a GOP. The frame with POC 8 is corrupted and estimated.

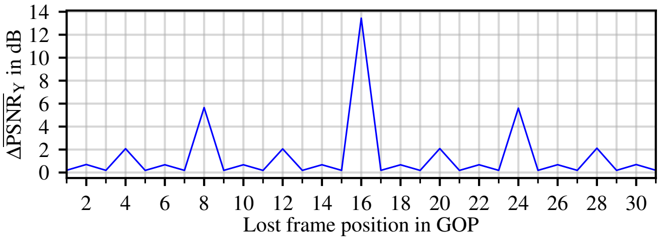


Fig. 4: Quality measurement $\Delta\overline{\mathrm{PSNR}}_{\mathrm{Y}}$ between error-free and error-concealed video plotted over the relative position of a lost frame in a GOP for NN.

II. Proposed Error Concealment Method

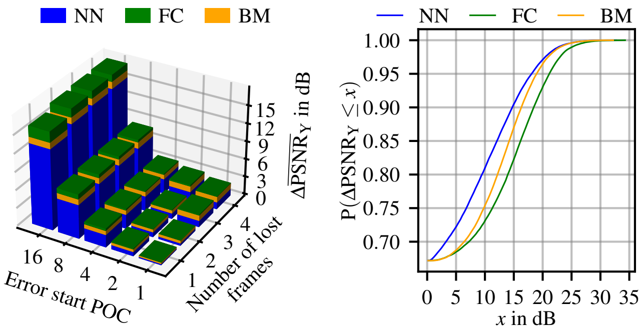
On the transmitter side, the VTM video encoder compresses the input video and delivers a bitstream or Network Abstraction Layer (NAL) unit stream to the channel encoder and communication system blocks. The channel encoder intelligently adds redundancy to the bitstream to increase its robustness against errors. These encoded bits are conveyed over an error prone channel. On the receiver side, the forward error correction (FEC) block recovers the bitstream from the encoded bits by removing the redundancy added at the transmitter side while it detects and corrects the errors added by the channel. Afterwards, the bitstream is passed to the VTM video decoder. Figure 1 shows a simplified block diagram of our EC solution integrated to the VTM video decoder. The CABAC block maps the received bitstream into syntax elements which after the inverse transform and quantizer are added to the prediction values. The prediction block contains the inter and intra prediction and the buffer block holds the reference frames needed for inter-prediction. For frames without uncorrectable errors, the switch is in the decoded frame (df) position.

FEC triggers the EC algorithm when an uncorrectable error is detected in the bit stream (dashed line). In this paper, we configure the coding structure as shown in Figure 2. Each GOP contains 32 frames with the 32nd frame always being an I-frame. All other frames within a GOP are B-frames. A slice is configured to contain an entire frame, since this configuration offers the best bitrate trade-off between channel and source coding [18]. Therefore, a NAL unit contains a whole frame as well. Just one erroneous bit in a NAL unit is enough to prevent CABAC from recovering the syntax elements of an entire frame and thus the frame is considered to be lost. If an error is detected, the EC algorithm is started: The neural network (NN) estimates the lost frame from two already decoded frames from a lower hierarchy level as indicated by the arrows in Figure 2. One from the future and one from the past in relation to the lost frame. Also, the switch is changed to the concealed frame (cf) position, indicating that the lost frame will be replaced with the error-concealed frame in the output sequence and saved in the buffer block for the inter-prediction of the following frames. Since the error-concealed frame is stored in the buffer, it may also be used as an input for the estimation of another lost frame.

The NN model that estimates any lost B-frame employs the RIFE-Large architecture [15]. RIFE uses an intermediate flow network named IFNet to estimate the intermediate flows $F_{t\to0}$ and $F_{t\to1}$ from an intermediate frame at $t = 0.5$ given two input frames I_0 and I_1. The input frames I_0 and I_1 are then individually backward warped using the intermediate flows $F_{t\to0}$ and $F_{t\to1}$ resulting in the two coarsely reconstructed frames $\widehat{I}_{t\leftarrow0}$ and $\widehat{I}_{t\leftarrow1}$. To fuse the two coarsely reconstructed frames into one, RIFE estimates a fusion map M and a residual signal Δ using a context extraction network and a fusion network from I_0, I_1, $F_{t\to0}$, $F_{t\to1}$, $\widehat{I}_{t\leftarrow0}$ and $\widehat{I}_{t\leftarrow1}$. The estimated intermediate frame $\widehat{I}_t$ is finally calculated following

$$\widehat{I}_t = M \odot \widehat{I}_{t\leftarrow0} + (1 - M) \odot \widehat{I}_{t\leftarrow1} + \Delta. \qquad (1)$$

The model was trained using the BVI-DVC dataset [19] which contains 800 sequences with 64 frames each. The spatial resolution varies between 3840x2176 and 480x272. Due to GPU-memory limitations, the model was trained using a resolution of 480x272. To overcome this limitation and still enable the model to infer full-HD sequences, the dataset was preprocessed to ensure that the model learns scale-invariant features. The sequences with a resolution of 3840x2176 were down-scaled to 1920x1088 and afterwards the following preprocessing steps were performed independently: (a) All sequences were down-scaled to 480x272. (b) All sequences were split into non-overlapping 480x272 parts. (c) A central 480x272 section was cropped from all sequences. After this procedure, which also serves as a form of data augmentation, our training dataset contained 8600 sequences with 64 frames each. The RIFE model was originally trained by its authors using the Vimeo90K dataset [20] in [15]. In a preliminary evaluation, we found that by training the network with Vimeo90K leads to a PSNR loss of 0.15 dB for luma in comparison to training it with our augmented BVI-DVC dataset for resolutions of 1920x1080 while achieving similar results for lower resolutions. The model was trained over 30 epochs with the full training dataset using AdamW as optimizer and an initial learning rate of 0.0001. The learning rate was gradually decreased to 0 using cosine annealing.



(a)

(b)

Fig. 5: $\Delta\text{PSNR}_\text{Y}$ between error-free and error-concealed video for burst errors over number of lost frames and first POC of the burst error (a). Probability $\text{P}(\Delta\text{PSNR}_\text{Y} \leq x)$ that $\Delta\text{PSNR}_\text{Y}$ between error-free and error-concealed video is less or equal to x for an average frame if one evenly distributed bit error occurs in the GOP (b).

III. Simulation Results and Discussion

All simulations were performed according to the JVET common test conditions (CTC) [21]. Sequences of classes B, C, D and E were encoded with the unmodified VTM 12.0 with quantization parameter values $\text{QP} = \{22, 27, 32, 37, 42\}$. However, for the figures presented in this paper we use $\text{QP} = 22$. Class A was not included in the simulations due to GPU memory limitations and class F was not included since screen content is not present in the BVI-DVC dataset. The random access configuration of VTM is chosen with a GOP size of 32. One slice per frame is selected. The VTM decoder is extended with NN based EC capabilities as described in Section II. It should be noted that the input and output for NN are whole frames and not patches. Additionally to NN, we also implemented two additional well-known EC methods as reference. The first reference method is frame copy (FC), which conceals a lost frame by simply copying the closest previously decoded frame in picture order count (POC) order. The other reference method is block matching (BM) [13], where two already decoded frames from a lower hierarchy level are partitioned into blocks and matched to each other to estimate motion vectors that are then used to error-conceal the lost frame. The parameters of BM were set to block size 16x16, and search window size 64x64 following [13]. For the adaptive filter of BM, the parameters were selected using a grid search and set to a filter size of 3x3, filter threshold of 100 and a filter weight of 10.

TABLE I: Expected value for $\Delta\overline{\text{PSNR}}$ between error-free and error-concealed video GOPs for all sequences within a class and different QPs. An evenly distributed single bit error within a GOP is assumed and different amounts of bits allocated to the corresponding NAL units are taken into account.

Video class	QP	NN			BM			FC		
		Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
B	22	4.17	1.76	2.49	4.42	1.78	2.51	5.06	2.23	3.00
	27	4.15	1.63	2.31	4.40	1.66	2.32	5.11	2.11	2.82
	32	3.72	1.39	1.95	3.96	1.40	1.94	4.66	1.84	2.42
	37	3.20	1.06	1.55	3.43	1.05	1.54	4.11	1.41	1.96
	42	2.72	0.87	1.33	2.92	0.86	1.30	3.58	1.19	1.70
C	22	4.07	2.39	2.81	4.68	2.65	3.06	5.28	3.11	3.58
	27	3.84	2.22	2.59	4.44	2.44	2.81	5.08	2.92	3.34
	32	3.21	1.85	2.16	3.76	2.04	2.35	4.39	2.49	2.85
	37	2.59	1.51	1.76	3.10	1.66	1.91	3.71	2.07	2.37
	42	2.04	1.26	1.43	2.49	1.41	1.57	3.06	1.78	1.99
D	22	3.30	1.65	2.02	4.67	2.15	2.66	5.26	2.44	2.97
	27	2.94	1.40	1.75	4.31	1.80	2.30	5.05	2.12	2.68
	32	2.31	1.08	1.40	3.52	1.39	1.82	4.26	1.64	2.15
	37	1.69	0.81	1.08	2.71	1.01	1.37	3.32	1.19	1.63
	42	1.19	0.62	0.82	1.94	0.76	1.03	2.45	0.92	1.25
E	22	3.66	1.16	1.36	4.32	1.37	1.58	4.69	1.68	1.92
	27	3.54	1.02	1.18	4.15	1.20	1.37	4.51	1.50	1.69
	32	3.00	0.77	0.85	3.55	0.92	1.02	3.89	1.21	1.32
	37	2.14	0.44	0.51	2.69	0.55	0.62	3.00	0.76	0.87
	42	1.42	0.31	0.35	1.86	0.39	0.43	2.13	0.57	0.64

We evaluated the performance of NN and compared it with FC and BM. First we show the average PSNR for each frame position within a GOP, as shown in Figure 3. An error in the 8th frame was introduced to every GOP such that the frame is lost. Each EC method produces an estimated frame to replace the lost one. PSNR_Y is computed for each frame, then it is averaged over all frames belonging to the same relative frame position in a GOP resulting in $\overline{\text{PSNR}}_\text{Y}$ in Figure 3. As it can be seen in Figure 2, an error in the 8th frame (POC 8) should only affect the POCs 1 to 15. This can also be observed in Figure 3, where frames 1 to 15 suffer PSNR losses which increase the closer they get to frame 8. As expected, the lost frame 8 suffers the highest PSNR loss. As shown in the figure, NN outperforms BM and FC regarding $\overline{\text{PSNR}}_\text{Y}$ by up to 2.2 dB and 4.0 dB, respectively. Table I confirms this same tendency in more detail. It gives the expected value for the average PSNR difference between error-free decoded video and error-concealed video over all frames in a GOP and over all GOPs in every video of the CTC. In this setting, an evenly distributed single bit error occurs in the GOP causing a single slice and thus a single frame to be lost. NAL units of different slices within the GOP allocate vastly different amounts of bits and larger NAL units have a higher probability to get a random bit error. We took this into account by calculating a weighted sum with the weight being the probability that the lost frame can be lost. It can been seen that NN performs better than BM and that BM performs better than FC for all classes and QPs regarding $\Delta\overline{\mathrm{PSNR}}_{\mathrm{Y}}$. For the chrominance, NN and BM outperform FC for every class and QP, while NN outperforms BM for all classes except class B in low QP ranges. Averaged over all classes, $\Delta\overline{\mathrm{PSNR}}_{\mathrm{Y}}$ is 3.85 dB, 4.52 dB and 5.09 dB for NN, BM and FC for QP 22, respectively.

In Figure 4 the difference between frames of error-free and error-concealed videos is measured and averaged over all GOPs of all classes for NN. BM and FC are always slightly below NN, but not plotted since the figure would otherwise be hard to read. It can be observed that a loss of frame 16 causes the highest $\overline{\mathrm{PSNR}}_{Y}$ decrease followed by the rest of the frames. The results of Figure 4 show a symmetry that can be explained by the hierarchy levels in Figure 2. Frames within the same hierarchy level cause the same $\overline{\mathrm{PSNR}}_{\mathrm{Y}}$ loss. The higher the hierarchy level of a lost frame, the lower the $\overline{\mathrm{PSNR}}_{\mathrm{Y}}$ loss is. This is to be expected, since frames with a lower hierarchy level have more dependent frames. Moreover, the temporal distance to the reference frames used by the EC algorithm is longer for frames in lower hierarchy levels.

During the transmission of signals, errors often occur as burst errors which may affect multiple adjacent frames. We ran the same experiment as described for Figure 4 again with the difference that we took bursts errors with up to 4 erroneous frames into account. Figure 5(a) shows $\Delta\overline{\mathrm{PSNR}}_{\mathrm{Y}}$ for burst errors in the range 1 - 4 starting at frame positions 1, 2, 4, 8 and 16 which represent the hierarchy levels 5, 4, 3, 2 and 1, respectively. As discussed before, the hierarchy level of lost frames within the coding structure has a significant impact on the expected PSNR loss. This behavior is once more confirmed in Figure 5(a). Furthermore, it can be observed that the hierarchy level of the starting frame has a much higher impact on the PSNR loss than the burst size. This is partly caused by the chosen positions of the first frame in the burst errors. If a burst error would start at frame 15, the following frame in coding order would be frame 24. Frame 24 and 15 are in hierarchy level 2 and 5, respectively. Therefore, frame 24 has a higher impact on the PSNR loss than frame 15.

If one evenly distributed single bit error is added to a GOP, the probability P_{naff} that a randomly selected frame within that GOP is not affected by this error is

$$P_{\text{naff}} = \sum_{h=1}^{5} P_{\mathrm{H}}(h) \cdot \left(1 - \frac{N_{\text{aff}}(h)}{N}\right) = 0.67, \qquad (2)$$

where h is the hierarchy level, $P_{\mathrm{H}}(h)$ the probability that a frame in h has a bit error, $N_{aff}(h)$ the number of dependent frames for h in the coding structure and $N = 31$ the total number of frames within a GOP excluding the I-frame. That means that the probability for a randomly selected frame to have zero PSNR loss within an erroneous GOP is $P_{\text{naff}} = 0.67$. The probabilities for PSNR losses higher than zero can be seen in Figure 5(b). As expected, all graphs start at 0.67, but the graph for NN shows higher probabilities for the same $\Delta\mathrm{PSNR}_{\mathrm{Y}}$, e.g. $\mathrm{P}(\Delta\mathrm{PSNR}_{\mathrm{Y}} \leq 10dB)$ is 0.81, 0.76 and 0.73 for NN, BM and FC, respectively. Figure 6 shows frames generated by all three EC methods. It can be observed that the frames error-concealed using NN are sharper than those using BM.

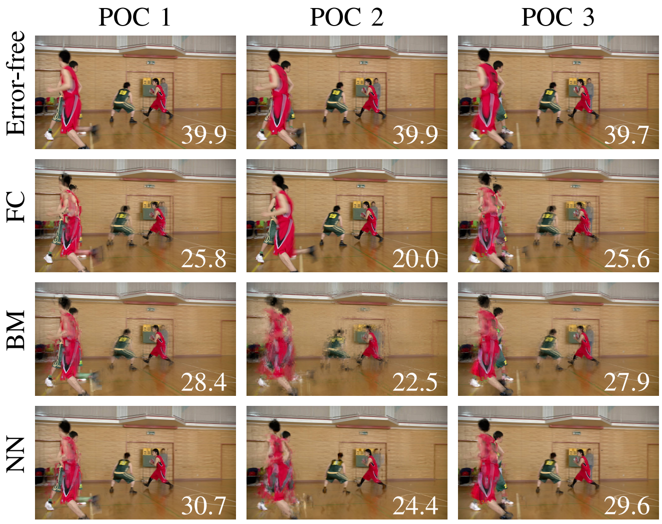


Fig. 6: The top row shows error-free decoded frames of the BasketballDrive sequence. In the other rows, POC 2 is estimated using NN, BM and FC approach. POC 1 and 3 are encoded with POC 2 in their reference picture lists. In each frame its corresponding $\mathrm{PSNR}_{\mathrm{Y}}$ is given.

Both EC algorithms increase the decoder time complexity for error-concealed frames relative to non error-concealed frames. Using one CPU core, the complexity increases by a factor of 251 for BM and 536 for NN. The complexity increase for NN drops to a factor of 8.5 when a GPU is used.

IV. Conclusion

This paper presents a neural network-based EC algorithm for VVC by estimating a lost B-frame from two successfully decoded, temporally neighboring frames. Its performance was evaluated for video communication systems over error prone channels using the CTC for neural network-based video coding. We implemented our proposed method NN and the reference methods BM and FC in the reference software VTM 12.0. NN outperforms BM and FC for all video classes in terms of PSNR measurements. The probability that the PSNR loss of a frame is lower than 10 dB increased by 7% from 0.76 for BM to 0.81 for NN. Compared to FC, the probability increased by 11% from 0.73. The expected PSNR loss value for a GOP with a single frame error decreased by 15% from 4.52 dB for BM to 3.85 dB for NN and by 24% from 5.09 dB for FC. To our knowledge, NN is currently the best performing EC algorithm for lost B-frames, which makes it a viable option as an EC solution for VVC.

REFERENCES

[1] B. Bross, J. Chen, S. Liu, and Y. Wang, "Versatile video coding (draft 10)," *ITU-T and ISO/IEC JVET-S2001*, 2020.

[2] G. Forecast, "Cisco visual networking index: global mobile data traffic forecast update, 2017–2022," *Update*, vol. 2017, p. 2022, 2019.

[3] Cisco, "Cisco annual internet report (2018–2023) white paper," 2020.

[4] 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.

[5] Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[6] J. Suh and Y. Ho, "Error concealment based on directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 295–302, 1997.

[7] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 1993, pp. 417–420 vol.5.

[8] M. Usman, X. He, M. Xu, and K. M. Lam, "Survey of error concealment techniques: Research directions and open issues," in *2015 Picture Coding Symposium (PCS)*, May 2015, pp. 233–238.

[9] C. Liu, R. Ma, and Z. Zhang, "Error concealment for whole frame loss in hevc," in *Advances on Digital Television and Wireless Multimedia Communications*, W. Zhang, X. Yang, Z. Xu, P. An, Q. Liu, and Y. Lu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 271–277.

[10] Y. Chang, Y. A. Reznik, Z. Chen, and P. C. Cosman, "Motion compensated error concealment for hevc based on block-merging and residual energy," in *2013 20th International Packet Video Workshop*, Dec 2013, pp. 1–6.

[11] T. Lin, N. Yang, R. Syu, C. Liao, and W. Tsai, "Error concealment algorithm for hevc coded video using block partition decisions," in *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, Aug 2013, pp. 1–5.

[12] Y. Zhang and Z. Li, "Multi-hypothesis-based error concealment for whole frame loss in hevc," in *MultiMedia Modeling*. Cham: Springer International Publishing, 2018, pp. 342–354.

[13] M. Usman, X. He, K.-M. Lam, M. Xu, S. M. M. Bokhari, and J. Chen, "Frame interpolation for cloud-based mobile video streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 831–839, 2016.

[14] W. Lotter, G. Kreiman, and D. Cox, "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning," in *International Conference on Learning Representations*, 2017.

[15] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "RIFE: Real-time intermediate flow estimation for video frame interpolation," *arXiv preprint arXiv:2011.06294*, 2020.

[16] A. Sankisa, A. Punjabi, and A. K. Katsaggelos, "Video error concealment using deep neural networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 380–384.

[17] M. Benjak, Y. Samayoa, and J. Ostermann, "Neural network based error concealment for VVC," in *Proceedings of the 28th IEEE International Conference on Image Processing (ICIP)*, Sep. 2021.

[18] Y. Samayoa and J. Ostermann, "Parameter selection for a video communication system based on hevc and channel coding," in *2020 IEEE Latin-American Conference on Communications (LATINCOM)*, Nov 2020, pp. 1–5.

[19] D. Ma, F. Zhang, and D. Bull, "BVI-DVC: A training database for deep video compression," *arXiv:2003.13552*, 2020.

[20] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision (IJCV)*, vol. 127, no. 8, pp. 1106–1125, 2019.

[21] S. Liu, A. Segall, E. Alshina, and R. Liao, "Jvet common test conditions and evaluation procedures for neural network-based video coding technology," *ITU-T and ISO/IEC JVET-T2006*, 2020.