Deep Learning based Spatial-Temporal In-loop filtering for Versatile Video Coding

Chi D. K. Pham[1], Chen Fu[1], and Jinjia Zhou[1,2]
[1] Hosei University, Tokyo, Japan
[2] JST, PRESTO, Saitama, Japan
{chi.kim.pham.do.94, chen.fu.6r}@stu.hosei.ac.jp, zhou@hosei.ac.jp

Abstract

The existing deep learning-based Versatile Video Coding (VVC) in-loop filtering (ILF) enhancement works mainly focus on learning the one-to-one mapping between the reconstructed and the original video frame, ignoring the potential resources at encoder and decoder. This work proposes a deep learning-based Spatial-Temporal In-Loop filtering (STILF) that takes advantage of the coding information to improve VVC in-loop filtering. Each CTU is filtered by VVC default in-loop filtering, self-enhancement Convolutional neural network (CNN) with CU map (SEC), and the reference-based enhancement CNN with the optical flow (REO). Bits indicating ILF mode are encoded under CABAC regular mode. Experimental results show that 3.78%, 6.34%, 6%, and 4.64% BD-rate reductions are obtained under All Intra, Low Delay P, Low Delay B, and Random Access configurations, respectively.

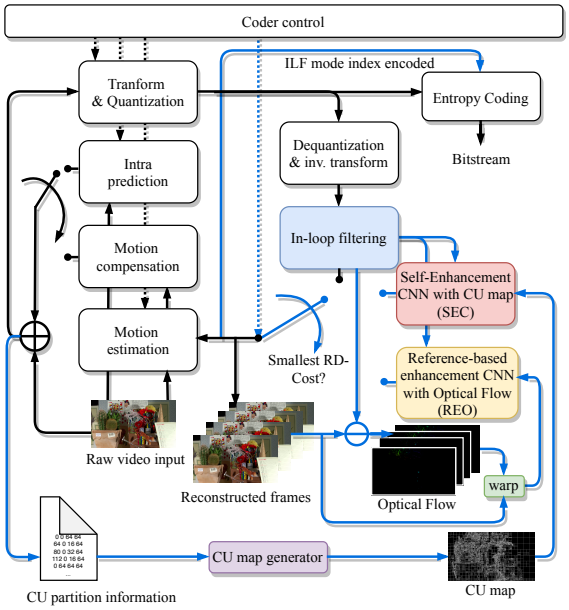

Figure 1. Illustration of the proposed STILF (blue arrows) integrated to VVC.

1. Introduction

The coming video coding standard Versatile Video Coding (VVC) [2] has exceeded the predecessor High Efficiency Video Coding (HEVC or H.265 [10]) in coding performance. Despite VVC in-loop filtering (ILF) improvement, reconstructed images are still affected by block-based coding and lossy compression which cause undesirable edges, blurring ringing artifacts, and missing information.

In recent years, Convolutional Neural Network (CNN) has significantly contributed to enhancing the VVC in-loop filters [4, 7, 8, 11, 6, 3]. In [4, 7], global skip connection is used for learning the residual between the image distorted by VVC encoding and the raw video frame. In [8, 3], skip connections have played a vital role in designing the dense residual convolutional neural network based in-loop filter (DRNLF) [8], and the dense residual convolutional neural network (DRN) [3] for enhancing VTM reconstructed images. Huang *et al.* [6] used Sobel and Laplace operators to generate the divergence and second derivative of reconstructed images, which highlight edge information and image details to improve the performance of residual learning. In [5], neighboring high-quality frames judged by a Peak Quality Frames detector are adopted for enhancing the reconstructed frames.

In this work, we propose a deep learning-based spatial-temporal in-loop filtering (STILF) - a framework that takes advantage of more potential resources in the encoder and decoder for enhancing the VVC reconstructed images to coding tree unit (CTU) precision. For each CTU, a syntax filtering mode compression chooses one of three different ILF modes: VVC default ILF, the proposed Self-enhancement CNN with CU map (SEC) exploiting the spatial information within the frame, or the Reference-based enhancement CNN with the optical flow (REO) utilizing the temporal correlations between frames. Compared to VVC

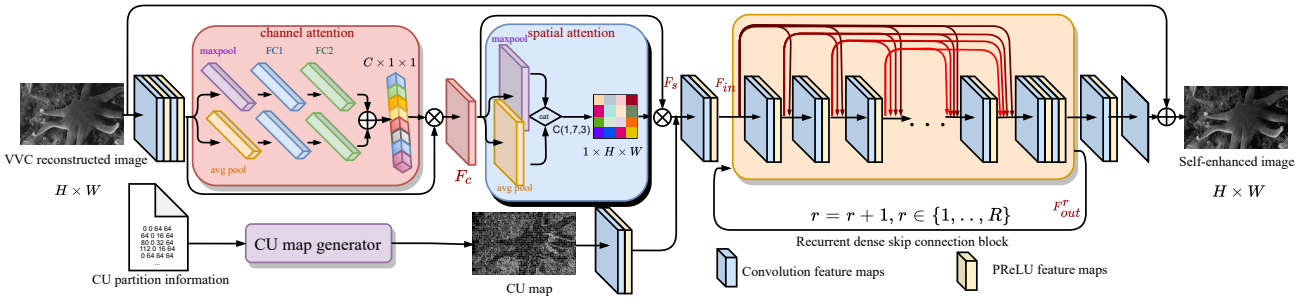

Figure 2. Self-enhancement CNN with CU map architecture (SEC). $C(k, s, p)$ indicates the convolution layer with k $s \times s$ kernels and padding p. Our RDSC block does recurrent for R times, output at time $r \in R$ are concatenated with the input F_{in} for the next loop $r + 1$.

Test Model (VTM) 9.3, we obtain 3.78%, 6.34%, 6%, and 4.46% BD-rate reductions under All Intra, Low Delay P, Low Delay B, and Random Access configurations, respectively. Compared to the anchor VTM 3.0, we obtain 4.0%-6.49% BD-rate reductions, which succeed 2.17%-3.29% BD-rate reductions performed by related works under various coding configurations.

2. The proposed STILF: deep learning based spatial-temporal in-loop filtering

Figure 1 shows the proposed STILF (blue arrows) integrated into VVC. The reconstructed frame is first processed by VVC ILF before being filtered by our self-enhancement CNN with CU map and the Reference-based enhancement CNN with optical flow. At the encoder, we calculate Rate-distortion (RD) cost of filtered images at the CTU level. For each CTU, an ILF mode is chosen if it performs the smallest RD-cost. Later, bits indicating chosen ILF mode is encoded by CABAC regular mode. At the decoder, CTU will be filtered by the ILF mode corresponding to the decoded bits.

2.1. Self-enhancement with CU Map (SEC)

Fig. 2 shows SEC architecture. Let $C(k, s, p)$ denote the convolution layer with k kernels size $s \times s$ and padding of p. At first, two convolution layers followed by the PReLU layer sequentially extract the feature maps from the input image. We use three self-attention mechanisms during the feeding forward: channel attention and spatial attention [12] for emphasizing useful information, and a feedback mechanism to take full advantage of the low-level and the high-level features. In SEC, the CU map from the VVC encoder also plays an important role: visualizing the possible blocking artifacts after encoding. The CU map can be visualized as a binary matrix where entries at CU borders are set to one, and the rest areas are zero.

Recurrent dense skip connection block. Our recurrent dense skip connection block (RDSC) block includes n convolutional layers, except the final layer, layer l^{th} takes $(l+1)n_f$ feature maps from $l-1$ previous convolution layers and RDSC input. High-level features are also fed back to the low-level layer for the next enhancement step. During feedforward, RDSC does the feedback for R times. The output F^r_{out} at loop $r^{th}, r \in \{1, ..., R\}$ of RDSC block is concatenated with input F_{in} feature maps to be the input for the next loop $(r + 1)^{th}$. Since loop $r = 1$ has no feedback F^{r-1}_{out}, the input feature maps are then duplicated and concatenated with themselves to be the input of the RDSC block. For SEC, n is set as seven, and the network does the recurrent R of four times. During training, we minimize the $L1$ loss between the ground-truth y and each network output $\hat{y}_r$:

$$L(\Theta) = \frac{1}{N.R} \sum_{i=1}^{N} \sum_{r=1}^{R} I^r \parallel y_0^i - \hat{y}_r^i \parallel_1 \quad (1)$$

where N is the number of training samples, and Θ denotes the learned network parameters. I^r indicates the weight of output $\hat{y}_r$ and is set I^r to one for all the outputs follow [13].

2.2. Reference-based Enhancement with Optical Flow (REO)

Fig. 3 shows the unfolding Reference-based enhancement CNN with Optical flow architecture. Since motion vectors between the current frame and the reference frame are not always available, it is better to calculate optical flow between the reconstructed reference frame and the reconstructed current frame. Our hypothesis is that if there is an optical flow between the current reconstructed image and the reconstructed reference image, it is also the optical flow between the enhanced current image and the enhanced reference image. SEC enhanced image will replace an input of REO if the reference image of that input is not activated in VVC. In REO, paddings are added during convolution to keep the size of input images. In each loop r^{th}, enhanced features output from the RDSC block are stored and concatenated with the features from the previous layer before input to the next loop $(r+1)^{th}$ of the recurrent RDSC block. For reference-based enhancement CNN, we set the number of convolution layers in RDSC block $n = 9$. R is set as five and the loss function can also be written as equation 1.

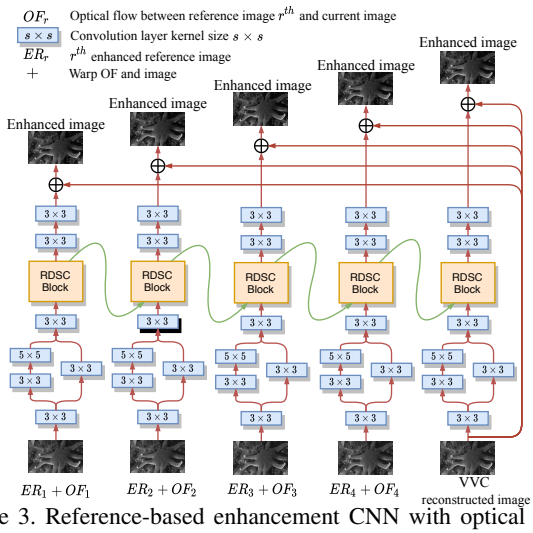

Figure 3. Reference-based enhancement CNN with optical flow (REO) architecture after unfolding. The order r of reference image is the from the long-term to short-term, POC after to before the current image in the reference picture sets List 0 and List 1.

3. Experiments

Experimental settings. Xiph.org sequences [9], which are not in the test set, are encoded by VVC Test Model (VTM) 9.3 under Low Delay P configuration. In testing, we evaluate STILF on VVC standard test sequences under common test conditions (CTC) [1] with the support of Tesla V100 GPUs. Bjøtegaard-Delta bit-rate (BD-rate) metric is used to evaluate the coding efficiency between different coding methods. For reference-based enhancement CNN, optical flows between the reference and the current frames are acquired by Lucas–Kanade method. The network parameters of SEC and REO are 1.7 $\times 10^6$ and 7×10^5, respectively.

Coding performance. Table 1 shows the overall performance of the proposed STILF. The work mainly focuses on enhancing the Y component, so the BD-rate of the U and V components slightly increases for some sequences. We obtain the Y BD-rate reduction of 3.78%, 6.34%, 6%, and 4.64% on the AI, LDP, LDB, and RA configurations. Besides, we recorded an encoding time of 1.31-1.84 and a decoding time of 8.69-73.03 on average CTC.

Abaltion study. We perform an ablation study on attention mechanisms, including self-attention (SA) and CU map (Table 2). The results show 1.84-11% coding performance has been reduced when cutting these mechanisms. In order to evaluate the performance of SEO and REC, an ablation study on each is performed. As a result, STILF with SEO has performed 5.0%, 0.5%, and 0.6% on Y, U, and V BD-rate reductions on LDP configuration. On the other hand, STILF with REC has performed 1.8%, -0.2%, and 0.3% Y, U, V BD-rate reduction under LDP configuration.

Subjective visual quality. Figure 4 illustrates the visual quality comparison of our proposal and the anchor VTM 9.3. It can be seen that STILF removes ringing artifacts and obtains better visual quality at lower bitrates than VTM 9.3.

Table 1. BD-rate(%) of our proposal compared to VVC under AI, LDP, LDB, and RA configurations. (Anchor: VTM 9.3)

Class	All Intra			Low Delay P		
	Y	U	V	Y	U	V
A1	-2.78	0.1	0.11	-4.15	-0.29	-0.06
A2	-2.2	0.03	0.03	-3.38	0.23	-0.02
B	-3.1	0.15	0.15	-5.21	-0.5	-0.11
C	-4.27	0.38	0.38	-6.79	-0.88	-0.47
D	-4.97	1.02	1.02	-9.81	-0.79	-0.53
E	-5.25	0.31	0.32	-8.14	0.61	-0.8
All	-3.78	0.35	0.35	-6.34	-0.34	-0.33
Class	Low Delay B			Random Access		
	Y	U	V	Y	U	V
A1	-2.95	-0.22	-0.21	-2.74	0.26	0.46
A2	-3.31	0.12	0.13	-3.51	-0.13	-0.05
B	-5.27	-0.39	-0.77	-3.95	0.45	0.37
C	-6.53	-0.54	-0.53	-4.98	-0.32	0.33
D	-9.44	-1.05	-1.07	-6.42	-0.47	0.05
E	-7.66	-0.54	-0.04	-5.95	0.13	0.17
All	-6	-0.46	-0.48	-4.64	-0.01	0.23

Table 2. BD-rate (%) results of the proposed STILF without self attention (SA) mechanisms and CU map under LDP configurations. (Anchor: VTM 9.3)

Class	STILF w/o SA	STILF w/o CU map	STILF w/o SA & CU map
B	-4.95	-5.15	-4.71
C	-6.68	-6.75	-6.12
D	-9.75	-9.79	-8.77
E	-7.81	-7.58	-6.92

Comparing with related works. For a fair comparison, we re-implement our proposal on VTM 3.0, where the works [4, 7, 8, 6] are performed. Table 3 shows our better performance than related works in AI, LDB, and RA configurations. STILF obtains coding gains of 6.49% and 5.43% Y BD-rate reductions, while the related works [4, 7, 8] obtains up to 2.17% and 2.23% BD-rate reductions on the Y component under LDB and RA configurations.

4. Conclusion

In this work, we propose a deep learning-based Spatial-Temporal In-Loop Filtering for the coming video coding standard VVC. Different from the existing approaches, this work takes advantage of more potential resources from the encoder and the decoder to improve the performance of CNN in-loop filters. By choosing the best in-loop filtering for each CTU, local areas are well enhanced and lead to quality improvement over the entire video frame. Consequently, 3.78% - 6.34% BD-rate reductions are obtained under various configurations.

Acknowledgement. This work is supported by JST, PRESTO Grant Number JPMJPR1757 Japan. We would

Table 3. BD-rate (%) measurement of our proposal (STILF) compared to the related works (anchor VTM 3.0).

Sequences	All Intra					Low Delay B				Random Access			
	[4]	[7]	[8]	[6]*	Ours	[4]	[7]	[8]	Ours	[4]	[7]	[8]	Ours
Tango2	-0.06	-0.75	-1.18	-	-2.6	-	-	-	-3.82	-0.04	-1.17	-1.67	-3.73
FoodMarket4	-0.14	-1.07	-1.7	-	-3.87	-	-	-	-3.49	-0.17	-0.83	-1.09	-3.59
Campfire	-0.04	-0.41	-1.74	-	-2.21	-	-	-	-2.46	-0.13	-0.9	-5.13	-2.63
CatRobot1	-1.08	-1.94	-2.52	-	-3.2	-	-	-	-5.21	-1.1	-2.47	-3.39	-5.08
DaylightRoad2	-0.2	-1	-1.33	-	-2.03	-	-	-	-5.19	-0.42	-2.46	-3.22	-5.48
ParkRunning3	-0.67	-1.41	-1.61	-	-1.52	-	-	-	-1.98	-0.5	-1.51	-1.14	-1.86
MarketPlace	-1.02	-1.36	-1.49	-	-2.42	-3.14	-0.73	-1.29	-4.01	-0.77	-1.18	-1.21	-3.22
RitualDance	-0.99	-2.02	-2.91	-	-4.38	-0.4	-1.57	-1.7	-4.34	-0.45	-1.92	-1.97	-3.68
Cactus	-0.62	-1.26	-0.8	-	-3.19	-0.31	2.45	-1.21	-5.72	-0.9	-1.52	-1.26	-4.92
BasketballDrive	-0.07	-0.66	-0.58	-	-3.58	-0.1	-0.91	-1.55	-5.34	-0.07	-1.04	-1.4	-4.91
BQTerrace	-0.44	-0.82	-0.86	-	-3.38	-0.11	-0.83	-1.91	-7.66	-0.67	-2.36	-2.52	-7
BasketballDrill	-2.7	-3.08	-3.85	-	-6.97	0.1	0.66	-0.88	-7.97	-1.16	-2.68	-1.77	-7.02
BQMall	-1.61	-2.43	-3.23	-	-4.81	-1	-1.7	-2.32	-8.76	-1.16	-2.61	-2.42	-6.93
PartyScene	-1.49	-1.94	-2.45	-	-3.8	-0.68	-0.57	-1.88	-7.04	-0.88	-2.11	-1.68	-5.39
RaceHorses	-1.08	-1.56	-1.85	-	-3.21	-0.87	-2.89	-2.3	-3.92	-0.73	-2.56	-2.42	-3.27
BasketballPass	-2	-2.64	-3.48	-	-5.1	-0.92	-3.93	-2.94	-8.67	-0.83	-3.57	-2.92	-6.42
BQSquare	-2.04	-2.73	-3.77	-	-7.3	-0.47	0.03	-1.93	-16.62	-0.96	-2.32	-2.05	-11.55
BlowingBubbles	-1.9	-2.28	-2.91	-	-3.86	-0.73	-1.15	-1.51	-6.54	-1.04	-2.2	-1.51	-5.85
RaceHorses	-2.77	-3.16	-4.13	-	-3.67	-2.29	-4.81	-3.63	-6.11	-1.68	-3.96	-3.54	-4.6
FourPeople	-2.24	-3.03	-4.13	-	-5.46	-1.02	0.57	-3.58	-8.62	-	-	-	-7.12
Johnny	-1.01	-2.04	-3.13	-	-5.93	-0.43	-0.07	-2.79	-11.17	-	-	-	-9.3
KristenAndSara	-1.77	-2.99	-3.68	-	-5.43	-1.16	-0.3	-3.27	-8.25	-	-	-	-5.85
Average A1	-0.08	-0.74	-1.54	-1.1	-2.89	-	-	-	-3.26	-0.11	-0.97	-2.63	-3.31
Average A2	-0.65	-1.45	-1.82	-1.94	-2.25	-	-	-	-4.13	-0.68	-2.15	-2.58	-4.14
Average B	-0.63	-1.22	-1.33	-2.51	-3.39	-0.81	-0.32	-1.53	-5.41	-0.57	-1.6	-1.67	-4.75
Average C	-1.72	-2.25	-2.85	-4.03	-4.7	-0.61	-1.13	-1.84	-6.92	-0.98	-2.49	-2.07	-5.65
Average D	-2.18	-2.7	-3.57	-5.33	-4.98	-1.1	-2.47	-2.5	-9.49	-1.13	-3.02	-2.5	-7.1
Average E	-1.68	-2.69	-3.65	-4.41	-5.61	-0.87	0.07	-3.21	-9.35	-	-	-	-7.42
Average All	-1.18	-1.84	-2.42	-3.29	-4.00	-0.85	-0.98	-2.17	-6.49	-0.72	-2.07	-2.23	-5.43

*: The work MGLNF [6] only provided average BD-rate results under AI configuration.

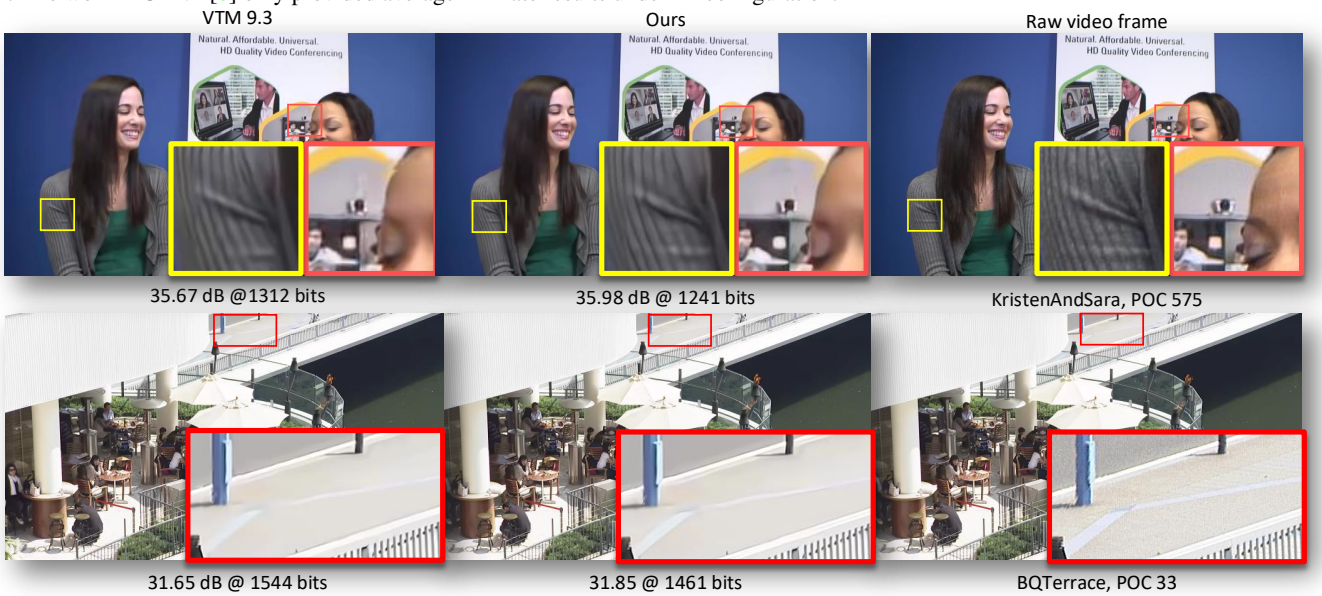

Figure 4. Visual comparison with VVC (anchor VTM 9.3). From top-down, KristenAndSara POC 575 and BQTerrace POC 33 encoded under Low Delay P with QP 37 are chosen for illustration.

like to thank Rikken for their kind sharing of powerful GPU servers.

References

[1] J. Boyce, K. Suehring, X. Li, and V. Seregin. JVET common test conditions and software reference configurations. *document Rep. JVET-J1010, San Diego, USA*, 2018. 3

[2] B. Bross, J. Chen, S. Liu, and Y.-K. Wang. Versatile Video Coding (Draft 10). *document Rep. JVET-S2001, Teleconference*, Apr. 2020. 1

[3] S. Chen, Z. Chen, Y. Wang, and S. Liu. In-loop filter with dense residual convolutional neural network for VVC. In *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 149–152. IEEE, 2020. 1

[4] Y. Dai, D. Liu, Y. Li, and F. Wu. AHG9: CNN-based in-loop filter proposed by USTC. In *document JVET-M0510, 13th JVET meeting*, 2019. 1, 3, 4

[5] Z. Guan, Q. Xing, M. Xu, R. Yang, T. Liu, and Z. Wang. Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 1

[6] Z. Huang, Y. Li, and J. Sun. Multi-Gradient Convolutional Neural Network Based In-Loop Filter For VVC. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. 1, 3, 4

[7] S. Naito K. Kawamura. A result of convolutional neural network filter. *document Rep. JVET-M0872, Marrakech, MA, USA*, Jan. 2019. 1, 3, 4

[8] Y. Li, L. Zhao, S. Liu, Y. Wang, Z. Chen, and X. Li. Test results of dense residual convolutional neural network based in-loop filter. *document Rep. JVET-M0508, Marrakech, MA, USA*, Jan. 2019. 1, 3, 4

[9] C. Montgomery et al. Xiph. org video test media (derf's collection), the xiph open source community, 1994. *Online, https://media.xiph.org/video/derf*. 3

[10] 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):1649–1668, 2012. 1

[11] M.Z. Wang, S. Wan, H. Gong, and M.Y. Ma. Attention-based dual-scale CNN in-loop filter for Versatile Video Coding. *IEEE Access*, 7:145214–145226, 2019. 1

[12] S. Woo, J. Park, J.Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[13] L. Zhen, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu. Feedback network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2