ON INTRA VIDEO CODING AND IN-LOOP FILTERING FOR NEURAL OBJECT DETECTION NETWORKS

Kristian Fischer, Christian Herglotz, and André Kaup

Multimedia Communications and Signal Processing
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Cauerstr. 7, 91058 Erlangen, Germany
{Kristian.Fischer, Christian.Herglotz, Andre.Kaup}@fau.de

ABSTRACT

Classical video coding for satisfying humans as the final user is a widely investigated field of studies for visual content, and common video codecs are all optimized for the human visual system (HVS). But are the assumptions and optimizations also valid when the compressed video stream is analyzed by a machine? To answer this question, we compared the performance of two state-of-the-art neural detection networks when being fed with deteriorated input images coded with HEVC and VVC in an autonomous driving scenario using intra coding. Additionally, the impact of the three VVC in-loop filters when coding images for a neural network is examined. The results are compared using the mean average precision metric to evaluate the object detection performance for the compressed inputs. Throughout these tests, we found that the Bjøntegaard Delta Rate savings with respect to PSNR of 22.2 % using VVC instead of HEVC cannot be reached when coding for object detection networks with only 13.6 % in the best case. Besides, it is shown that disabling the VVC in-loop filters SAO and ALF results in bitrate savings of 6.4 % compared to the standard VTM at the same mean average precision.

Index Terms— machine to machine communication, video coding for machines (VCM), neural object detection, versatile video coding (VVC), in-loop filtering

1. INTRODUCTION

Thanks to their outstanding performances, more and more applications are using neural networks that solve different tasks on multimedia data. However, these neural networks have the major drawback that they all need high-end graphic processing units (GPUs) which are first expensive and second not practical to battery driven devices like cars or unmanned aerial vehicles (UAVs). Thus, it is required to outsource the neural network execution to a remote server that is equipped with a GPU. However, the rate-constrained channels demand for compressing the multimedia data. Therefore, we investigate whether classical video coding for humans, as depicted in Fig. 1, also operates effectively when the decoded video is analyzed by a neural network detector as shown in Fig. 2.

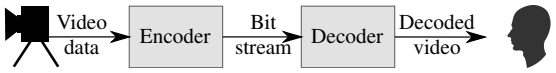

Fig. 1: Video coding for humans.

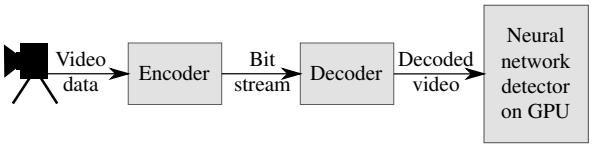

Fig. 2: Video coding for machines.

Coding input data for neural networks can be attributed to the field of *Video Coding for Machines (VCM)* that deals with machine to machine (M2M) communication. According to the Cisco Visual Networking Index [1], 51 % of total devices and connections will be used for M2M communication systems in 2022, and the amount of M2M internet traffic will increase from 4 % in 2017 to 7 % in 2022 , so it is reasonable to put effort in coding M2M data. The significance of this topic is also underlined by the fact that MPEG introduced an ad hoc group on VCM in 2019 [2], which tackles several use cases like smart factory, video surveillance, and autonomous vehicles. In the scope of this paper we focus on the latter one.

One use case demanding for VCM is a collision avoidance system for an UAV as presented in [3]. There, the captured video data has to be sent to a cloud server that runs the neural detection network, since there is no place for the heavy and energy demanding GPU on the drone.

Several works already exist in the field of compressing data for neural networks. In [4], the authors proposed a modification of the video coding standard H.264 that preserves important areas, while background pixels are compressed stronger. Choi and Bajić proposed a new rate control method for H.265 that is adapted to object detection and spends more bits on areas that are vital for the used YOLO9000 [5] object

The authors gratefully acknowledge that this work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under contract number KA 926/10-1.

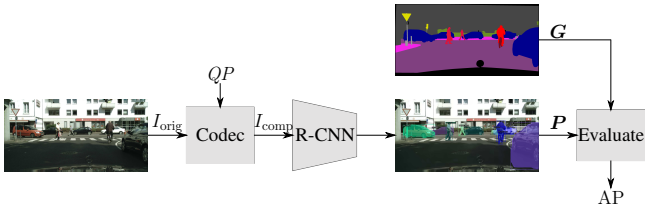

Fig. 3: Signal flow investigating the codec performance for R-CNNs.

Table 1: *Cityscapes* object categories with the number of instances in validation set with 'MC' and 'BC' standing for motorcycle and bicycle, respectively.

Person	Rider	Car	Truck	Bus	Train	MC	BC
3399	544	4656	93	98	23	149	1169

detection network [6]. Another work improves the object detection performance by creating saliency maps that helps the video encoder to focus on the relevant image parts [7].

More theoretical work has been done in [8] where investigations were performed on the influence of blur, noise, and JPEG compression to neural classification networks. The same authors also found that humans perform better in solving classification tasks on noisy and blurred images than deep neural networks [9]. In [10], a method for CNNs was designed to be more robust against noisy, blurred, or JPEG compressed images.

The investigations of this paper are twofold. First, we compare the performances of the High Efficiency Video Coding (HEVC) [11] with its successor Versatile Video Coding (VVC) [12] for intra coded frames in an autonomous driving scenario and two different region-based convolutional neural network (R-CNN) architectures. Second, we investigate the three VVC in-loop filters comparing their influence on the HVS and the R-CNNs for the given setup.

2. ANALYTICAL METHODS

In Fig. 3, the basic flow chart of our simulations is depicted. The input RGB image I_{orig}, which should be analyzed, is first transformed to the YCbCr color format with 4:2:0 subsampling and then coded with either HEVC or VVC, controlling the quality with the quantization parameter (QP). The resulting compressed image I_{comp} has to be back-converted to the RGB colorspace, before applying an R-CNN. Subsequently, the resulting predictions $\boldsymbol{P}$ from the network, which include the pixels belonging to the instance, the predicted class, and the certainty, are evaluated against the ground-truth (GT) data $\boldsymbol{G}$ with the average precision metric (AP).

2.1. Dataset

All simulations are conducted on the *Cityscapes* dataset [13], which includes automotive stereo data observing urban street scenes in Germany. From that dataset, the 500 uncompressed validation images of the left camera with a spatial resolution of 2048×1024 pixels are used. The GT data is labeled pixel-wise including multiple classes. For our purpose, we consider the eight classes listed in Table 1 with the number of instances occurring in the validation set.

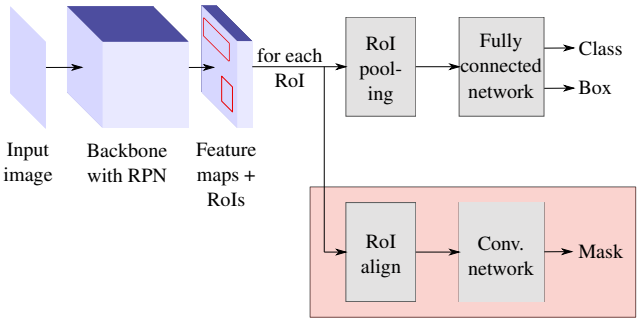

Fig. 4: Structure of used R-CNNs; Upper branch is used for Faster R-CNN, while the lower red branch is additionally used for Mask R-CNN to gain the pixel accurate masks.

For evaluating the object detection performance, the AP is used with the adaptations from the *Cityscapes* challenge as described in [13]. The AP value calculates the area under the precision-recall curve for several Intersection over Union (IoU) thresholds and for each object category. Thereby, AP_{50} is calculated for all predicted instances having an IoU with the GT instances above a threshold of 50 %, whereas AP is averaged over ten IoU thresholds from 50 % to 95 % in steps of 5 %. For the mean AP (mAP) metric, the AP values of all object categories are averaged. Two modifications are made compared to the original *Cityscapes* evaluation code for instance-level semantic labeling from [14]. For Faster R-CNN, the GT pixel-wise annotations are converted to bounding boxes for a comparable calculation of IoU values. Second, the mAP is calculated with a weighted average based on the number of instances of each class, to avoid that classes with a low instance count (e.g. train) contribute as much to the mAP and therewith to the coding performance evaluation as classes with many instances (e.g. car).

2.2. Investigated Codecs

In the scope of this paper, we used the HEVC test model (HM) software [15] in version 16.2 and the VVC test model (VTM) in version 6.0 for VVC [12]. We relinquish using traditional image codecs like JPEG, because they were shown to perform worse than HEVC on images [16].

Since the *Cityscapes* dataset only contains single images, the video codecs HEVC and its successor VVC are tested in the all-intra configuration.

2.3. Investigated Object Detection R-CNNs

The state-of-the-art neural object detection networks Faster R-CNN [17] and Mask R-CNN [18] were chosen to evaluate the influence of intra-frame compression on object detection.

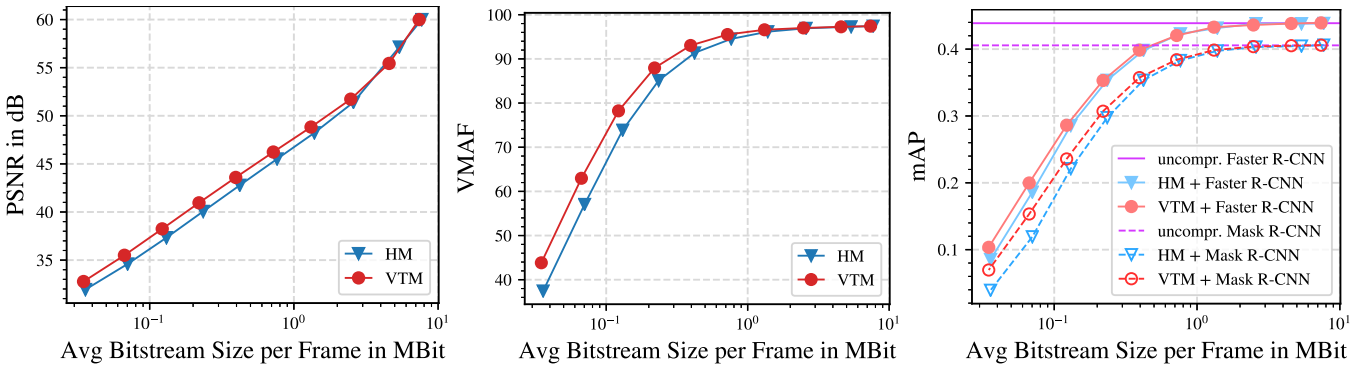

Fig. 5: Rate-distortion diagrams with respect to PSNR, VMAF, and mAP comparing HM 16.2 with VTM 6.0 for 500 *Cityscapes* validation images and all intra configuration for $QP \in \{2, 7, 12, 17, 22, 27, 32, 37, 42, 47\}$.

Faster R-CNN outputs a class label with a certainty score and a bounding box for each potential object as depicted in the upper branch of Fig. 4. To that extent, a feature map is extracted from the input image by a backbone consisting of convolutional layers that are adapted from classification networks. From this features the region proposal network (RPN) derives regions of interest (RoIs), which are then classified and refined by the subsequent fully connected layers.

Mask R-CNN is a derivative of the Faster R-CNN structure targeting the problem of semantic segmentation with per-pixel classification. To create the pixel-accurate masks, a parallel convolutional network is attached to the Faster R-CNN structure as shown in Fig. 4. More detailed information on both used networks can be found in [17] and [18].

We use the *PyTorch* implementations from the *Detectron2* library [19] for both R-CNNs. For Mask R-CNN, we take the already existing model that has been trained on the *Cityscapes* training set. Its backbone is a Residual Net with 50 layers (ResNet-50) [20] and Feature Pyramid Network (FPN) [21] structure. The same backbone is taken for Faster R-CNN, but since there is no pre-trained *Cityscapes* model available in [19], the already existing model trained on the COCO dataset is taken from [19] as initial weights. Subsequently, this model is further trained on the *Cityscapes* classes and training images, where the GT pixel masks have been converted to bounding boxes before, for 42000 iterations, a batch size of 7 images, and a learning rate of 0.00025.

3. EXPERIMENTAL RESULTS

3.1. Comparison HEVC vs. VVC

The performance of HM and VTM are compared by plotting peak signal-to-noise ratio (PSNR), Video Multi-Method Assessment Fusion (VMAF) with the *vmaf_v0.6.1.pkl* model [22] and mAP over the bitrate. These rate-distortion curves for QP values from 2 to 47 in steps of 5 can be found in Fig. 5.

For object detection, the mAP does not significantly decrease for input images compressed with a QP lower than 17 compared to the uncompressed input data. With higher QP than 17, the network precision is dropping constantly, where VMAF shows a similar course. Contrary, PSNR continues to increase with higher bitrates.

Table 2: BDR in % with respect to the particular quality metric using HM with the corresponding R-CNN as anchor for $QP \in \{22, 27, 32, 37\}$.

	PSNR	VMAF	mAP	mAP_{50}
Faster R-CNN	-22.17	-25.55	-6.01	-5.79
Mask R-CNN			-13.56	-11.24

In Table 2, Bjøntegaard Delta Rate (BDR) [23] values for $QP \in \{22, 27, 32, 37\}$ are listed as suggested by JVET [24]. Higher QP are not considered, since they result in mAP values that are too small for practical applications. The BDR compares two rate-distortion curves by calculating the average bitrate savings for the same quality between two codecs. Originally, PSNR is taken as quality metric for calculating the BDR, however, we also use VMAF, mAP, and mAP_{50} analogously as it has been done in [6]. Considering these BDR values, the coding gains between VTM and HM based on VMAF and PSNR above 22 % are superior to taking the mAP into account (up to 13.6 %). Hence, the optimizations made for the VTM are considerably more effective for the HVS than for neural detection networks, because the VTM is mainly optimized for delivering a decent PSNR value which does not necessarily results in a high detection rate. Since the coding artifacts have a higher influence on the task of pixel-wise segmentation than for drawing bounding boxes around the objects, the BDR savings for Mask R-CNN are higher than for Faster R-CNN. The coding gains for mAP are higher compared to mAP_{50}, since the mask or the bounding box has to fit more precisely for mAP and thus coding artifacts are more disturbing for object detection.

3.2. Influence of In-Loop Filters on R-CNNs

Derived from the previous results, additional investigations are taken whether existing VVC tools are actually improving the mAP when coding for R-CNNs. To that extent, the influence of the three VVC in-loop filters on object detection is

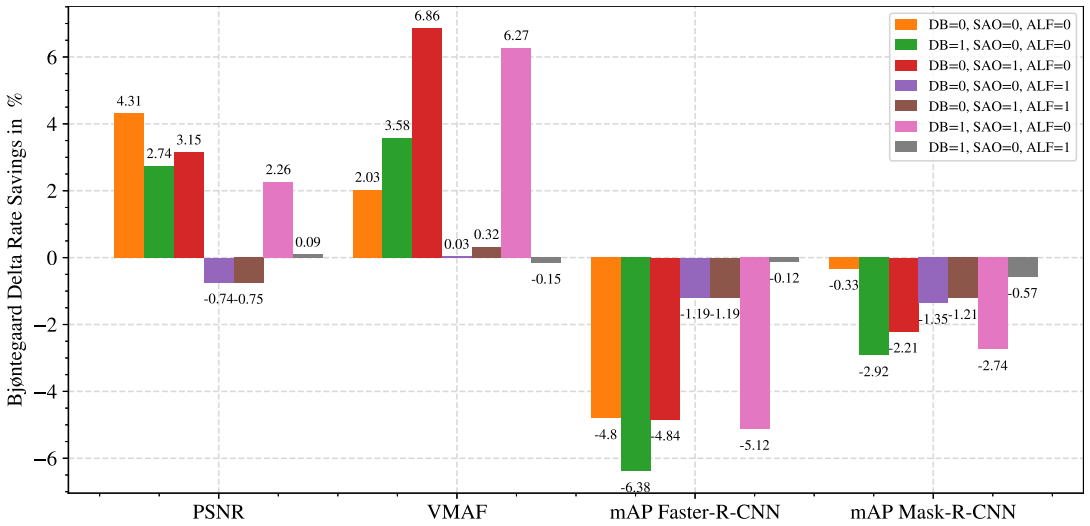

Fig. 6: BDR in % with respect to PSNR, VMAF, and mAP for Faster and Mask R-CNN comparing all possible permutations of activated or deactivated in-loop filters using the standard VTM with all in-loop filters activated as anchor; Used $QP \in \{22, 27, 32, 37\}$; Negative BDR values represent bitrate savings compared to the anchor.

compared with their influence on the HVS. The three in-loop filters are the de-blocking filter (DB) [25] minimizing blocking artifacts, sample adaptive offset filter (SAO) [26] categorizing pixels and additionally transmitting suitable offset values, and the adaptive loop filter (ALF) [27] convolving the output with a suitable filter as the last step of the coding chain.

In Fig. 6, the resulting BDR savings for each permutation of active in-loop filters for $QP \in \{22, 27, 32, 37\}$ are shown, taking the standard VTM with all in-loop filters activated as anchor. For BDR with respect to PSNR, deactivating in-loop filters results in decreased rate-distortion performance with respect to standard VTM for most permutations. However, there are two permutations with activated ALF and deactivated DB filter (purple and brown) that show minor bitrate savings compared to the standard VTM. Considering VMAF, deactivating the in-loop filters also results in higher bitrates at same VMAF and in the worst case 6.86 % more bitrate is required.

Taking the in-loop filter influence on BDR for Faster R-CNN into account, the opposite behavior can be observed. There, deactivating in-loop filters results in high bitrate savings at the same detection rate for Faster R-CNN. When only activating the DB filter (green), 6.38 % bitrate can be saved compared to the standard VTM. When using Mask R-CNN to evaluate the codec performance, the results are similar to Faster R-CNN since all permutations with at least one in-loop filter deactivated require less bitrate than the standard VTM. Again, only activating the DB filter results in the highest bitrate savings of 2.92 % at the same mAP.

These results indicate that the R-CNNs are mainly sensitive to blocking artifacts while other artifacts that are reduced by SAO and ALF do not decrease the detection rate significantly. Contrary to BDR with respect to PSNR or VMAF, additionally activating the SAO or the ALF to the DB filter has no positive effect on the detection quality and also requires extra bits that lowers the coding performance. The bitrate savings for Faster R-CNN are higher than for Mask R-CNN when deactivating SAO and ALF since the task of drawing bounding boxes is less prone to coding artifacts than pixel-wise segmentation and thus the in-loop filters have less influence. All in all, it can be recommended deactivating the SAO and ALF when coding data for neural detection networks.

4. CONCLUSION

Video coding for machines is a relevant topic, and rate-mAP curves for state-of-the-art codecs and R-CNNs were presented within this paper and compared against HVS metrics. The first experiment showed that the large BDR savings above 20 % comparing VVC against HEVC can only be observed for the metrics that represent the human as final user. For the case that R-CNNs analyze the coded image, the coding gains were not that high, although they are still between 5 % and 14 % depending on the used network and mAP metric. Furthermore, it was found that when deactivating all in-loop filters except the DB filter, significant bitrate savings can be achieved when coding for R-CNNs. Additionally, computational complexity can be saved omitting the SAO and the ALF. All in all, new VVC optimization methods have to be found when coding for neural networks in order to achieve bitrate savings in the same dimension as for PSNR and VMAF. Possible approaches might consider replacing the PSNR in the rate-distortion optimization of the VVC with metrics better reflecting the R-CNN behavior. Besides, it should be investigated whether the presented results are also valid for inter coding. However, this requires a suitable labeled and uncompressed video dataset.

5. REFERENCES

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017-2022," Tech. Rep., Feb. 2019.

[2] Y. Zhang and P. Dong, "MPEG-M49944: Report of the AhG on VCM," Tech. Rep., Moving Picture Experts Group (MPEG) of ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Oct. 2019.

[3] J. Lee, J. Wang, D. Crandall, S. Šabanović, and G. Fox, "Real-time, cloud-based object detection for unmanned aerial vehicles," in *Proc. IEEE International Conference on Robotic Computing (IRC)*, Apr. 2017, pp. 36–43.

[4] A. D. Bagdanov, M. Bertini, A. Del Bimbo, and L. Seidenari, "Adaptive video compression for video surveillance applications," in *Proc. IEEE International Symposium on Multimedia*, Dec. 2011, pp. 190–197.

[5] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.

[6] H. Choi and I. V. Bajic, "High efficiency compression for object detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 1792–1796.

[7] L. Galteri, M. Bertini, L. Seidenari, and A. Del Bimbo, "Video compression for object detection algorithms," in *Proc. International Conference on Pattern Recognition (ICPR)*, Aug. 2018, pp. 3007–3012.

[8] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *Proc. International Conference on Quality of Multimedia Experience (QoMEX)*, June 2016, pp. 1–6.

[9] S. Dodge and L. Karam, "A study and comparison of human and deep learning recognition performance under visual distortions," in *Proc. International Conference on Computer Communication and Networks (ICCCN)*, July 2017, pp. 1–7.

[10] S. Ghosh, R. Shet, P. Amon, A. Hutter, and A. Kaup, "Robustness of Deep Convolutional Neural Networks for Image Degradations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[11] 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, Dec. 2012.

[12] J. Chen, Y. Ye, and S. Kim, "JVET-O2002: Algorithm description for versatile video coding and test model 6 (VTM 6)," Tech. Rep., Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Sept. 2019.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] M. Cordts and M. Omran, "The cityscapes dataset," https://github.com/mcordts/cityscapesScripts, 2017.

[15] Joint Collaborative Team on Video Coding, "HEVC Test Model ReferenceSoftware (HM)," https://hevc.hhi.fraunhofer.de/.

[16] T. Nguyen and D. Marpe, "Performance analysis of HEVC-based intra coding for still image compression," in *Proc. Picture Coding Symposium (PCS)*, May 2012, pp. 233–236.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[18] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988.

[19] Y. Wu, A. Kirillov, F. Massa, W. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[21] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 936–944.

[22] Netflix Inc., "VMAF - video multi-method assessment fusion," https://github.com/Netflix/vmaf, 2016.

[23] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Proc. ITU-T Video Coding Experts Group (VCEG)*, Apr. 2001.

[24] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Shring, "JVET-N1010: JVET common test conditions and software reference configurations for SDR video," Tech. Rep., Joint Video Experts Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Mar. 2019.

[25] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.

[26] C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, C. Tsai, C. Hsu, S. Lei, J. Park, and W. Han, "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1755–1764, Dec. 2012.

[27] C.Y. Tsai, C. Chen, T. Yamakage, I. S. Chong, Y. Huang, C. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz, and S. Lei, "Adaptive loop filtering for video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 934–945, Dec. 2013.