

Luma Range Scaling for Enhanced VVC Efficiency in Video Coding for Machines

Tero Partanen¹, Alban Marie¹, Alexandre Mercat¹, Jarno Vanne¹,
Miska M. Hannuksela², Honglei Zhang², Alireza Aminlou², and Francesco Cricri²

¹Ultra Video Group, Tampere University, Finland

²Nokia Technologies, Finland

{tero.partanen, alban.marie, alexandre.mercat, jarno.vanne}@tuni.fi,

{miska.hannuksela, honglei.l.zhang, alireza.aminlou, francesco.cricri}@nokia.com

Abstract—Recent years have shown significant growth in video data traffic for machine vision applications, catalyzing new standardization efforts in video coding for machines (VCM). These activities focus on compressing images and videos for machine vision tasks, rather than for human viewing. In this work, we propose a novel method that scales down the luma range to enhance the coding efficiency of Versatile Video Coding (VVC) for machine consumption. This method results in a lower bitrate after encoding and has only minimal adverse effects on the accuracy of machine vision tasks. In our experiments, we down-scale the luma channel of the input video using luma-scaling factors from 0.2 to 0.9 and evaluate coding results with optional back-scaling to the original range before machine vision tasks. Our results with the VVC Test Model (VTM) demonstrate that the proposed technique achieves coding gain of up to 37.9% and 46.1% for the same object detection and tracking accuracy, respectively.

Keywords—Video Coding for Machines (VCM), Versatile Video Coding (VVC), Common Test Conditions (CTC), Machine Vision

I. INTRODUCTION

Application domains like surveillance, autonomous driving, intelligent transportation, and smart manufacturing are increasingly reliant on automated visual data analysis, which has led to the explosion of visual data traffic and the proliferation of machine vision applications. This trend has catalyzed new standardization activities within the *Moving Picture Experts Group (MPEG)* and *Joint Video Exploration Team (JVET)* in novel research field called *video coding for machines (VCM)* [1]. The objective of VCM is to develop advanced video coding techniques tailored specifically for machine-based or hybrid machine-human consumption.

The most noteworthy video coding standards, such as *High Efficiency Video Coding (HEVC)* [2] and *Versatile Video Coding (VVC)* [3], were initially developed to compress videos according to the characteristics of human visual system. Since these traditional coding approaches may be suboptimal for machine consumption, JVET has recently investigated various techniques to enhance VVC coding efficiency for machine vision tasks, including both pre- and post-processing methods [4].

In addition to JVET, numerous works have proposed to enhance the coding efficiency of traditional coding standards for

machine vision. A common approach is to use pre-processing techniques through *region of interest (ROI)*-based methods. These methods detect salient areas within video frames and utilize the saliency information in two primary ways: 1) by guiding the encoder to allocate more bits to salient areas than the other regions of the frame [5]–[7]; or 2) by modifying the video input by either blurring [8], [9] or completely removing non-salient areas [9], [10]. It is noteworthy that saliency detection can significantly increase the computational complexity of these methods. Moreover, a part of them are designed for machine-only video analysis. Alternatively, some non-ROI methods enhance coding efficiency by modifying the entire frame, using techniques such as spatial downsampling [11] or truncating the least significant bit of luma values for bit-depth reduction [12].

In this paper, we propose a method wherein the luma channel of the input video is scaled down, with a luma-scaling factor between 0 and 1, to a limited dynamic range prior to encoding. This technique effectively reduces the output bitrate of the VVC encoder, analogous to the traditional approach of increasing the *quantization parameter (QP)*. Our solution is made up of two steps: 1) a pre-processing step for luma down-scaling and 2) an optional post-processing step for luma back-scaling. According to our results, the proposed method offers several advantages: (i) significantly improved VVC coding efficiency for machine consumption; (ii) inherently low computational complexity overhead; (iii) broad applicability to both machine vision and human viewing scenarios; (iv) standard-compliance; and (v) encoder agnosticism. To the best of our knowledge, this is the first work that proposes and demonstrates the potential of luma range scaling for VCM.

The remainder of this paper is organized as follows. Section II reviews related pre-processing techniques in the literature. Section III details the proposed luma range scaling technique. Section IV describes the experimental setup used to validate the effectiveness of the proposed method, followed by comprehensive experiments and results analysis in Section V. Ultimately, Section VI concludes the paper.

II. RELATED WORKS

Numerous studies have investigated the advantages of pre-processing techniques aimed at improving video coding efficiency for machines. Typically, the optimization target is to

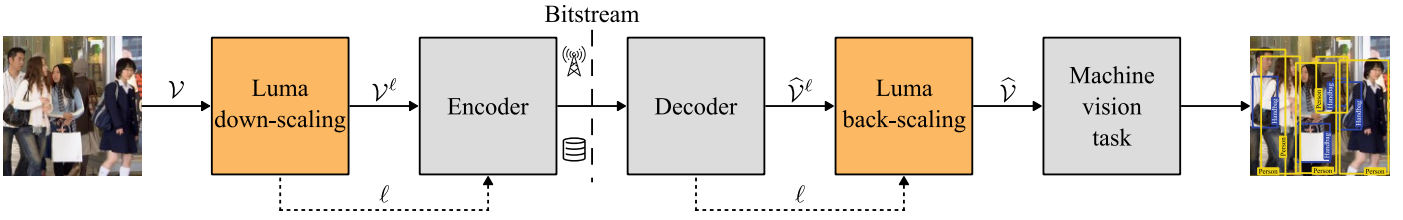


Fig. 1. Pipeline for the proposed luma range scaling technique. Added steps are highlighted in orange.

reduce the bitrate while minimizing the detrimental impact on the accuracy of the machine vision task. In general, techniques that introduce an additional pre-processing step can be classified into two distinct categories [4].

The first category encompasses pre-processing techniques that do not modify the source video but extract some high-level information to guide the subsequent encoding process. This approach is commonly referred to as saliency- or *region-of-interest (ROI)*-optimized video coding. For instance, Choi *et al.* [13] and Cai *et al.* [14] proposed methodologies to guide the encoder to allocate more bits to salient regions than non-salient regions of the video frames. The bit allocation is guided by importance maps generated from the convolutional layers of an object detection model. Similarly, Fischer *et al.* [15] presented an approach, where an object detection model is employed to detect objects treated as the ROI areas. The quality of these ROI areas is preserved, whereas the non-ROI areas undergo more aggressive compression.

The second category encompasses pre-processing techniques that intentionally modify the visual content prior to encoding. They typically involve blurring or complete removal [4] of non-ROI areas. For instance, Bagdanov *et al.* [8] proposed to blur the background, thus reducing the overall bit consumption. On the other hand, Aliouat *et al.* [10] introduced a method that completely removes the background prior to encoding for object-based machine vision tasks. Furthermore, Laitinen *et al.* [9] investigated both background blurring and removal in a unified multi-layer coding scheme for both human and machine consumption. In addition to ROI-based techniques, other methods that modify the input source have also demonstrated a potential to improve coding efficiency. For example, it has been shown that coding efficiency opportunities can be obtained from an appropriate selection of the spatial downsampling factor as shown by Marie *et al.* [11]. More recently, Ding *et al.* [12] improved the coding efficiency in object detection and tracking tasks merely by truncating the least significant bit in videos prior to encoding.

The existing pre-processing techniques demonstrate notable potential to achieve enhanced coding efficiency for machine vision tasks, but some of them also introduce inherent disadvantages. For instance, ROI-based techniques typically require the use of an additional ROI detection step, which may pose constraints for embedded cameras with limited computational resources. Additionally, the use of object detection modules to determine salient areas limits the advantages that VCM strives for, i.e., outsourcing the computation of machine vision tasks. Additionally, background manipulation provides a highly task-specific video bitstream,

which is typically also challenging for human analysis and interpretation.

The bit truncation [12] is most related to our proposal. However, the sole bit truncation technique is very coarse compared to our proposal. Additionally, their document has not undergone peer review, and there are notable limitations in their experimental setup that potentially limit the reliability of their findings, as discussed later in Section V.

III. PROPOSED LUMA RANGE SCALING

The proposed luma range scaling technique aims to enhance the coding efficiency for machine vision tasks. In essence, it *scales down* the luma channel to a reduced range before encoding. Such a pre-processing step decreases the bitrate required to encode the video while potentially having a less adverse effect on the accuracy of machine vision tasks compared to increasing the QP during the encoding process.

Fig. 1 illustrates the pipeline of the proposed luma range scaling technique. Let \mathcal{V} be a video composed of pixels $\mathbf{p} = (p_Y, p_U, p_V) \in \mathcal{V}$, where $p_Y, p_U,$ and p_V represent the luma and chroma channels in the $Y'CbCr$ colorspace. The video \mathcal{V} is first processed by the luma down-scaling step. Let $\ell \in]0; 1[$ be the luma-scaling factor. The luma down-scaling step consists of multiplying the luma p_Y , of each pixel \mathbf{p} within \mathcal{V} by ℓ as

$$\forall \mathbf{p} = (p_Y, p_U, p_V) \in \mathcal{V}, \\ \mathbf{p}^\ell = (p_Y^\ell, p_U^\ell, p_V^\ell) = (\lfloor \ell p_Y \rfloor, p_U, p_V), \quad (1)$$

where $\mathbf{p}^\ell = (p_Y^\ell, p_U^\ell, p_V^\ell) \in \mathcal{V}^\ell$ is the pixel that belongs to the luma down-scaled video \mathcal{V}^ℓ and $\lfloor \cdot \rfloor$ is an operator that rounds to the nearest integer. It can be observed that (1) becomes equivalent to the identity function when $\ell = 1$. As the luma-scaling factor converges towards zero, i.e., $\ell \rightarrow 0$, (1) essentially affects the distribution of luma pixel values p_Y^ℓ in \mathcal{V}^ℓ by down-scaling them towards smaller values. Subsequently, a reconstruction of the luma down-scaled video $\hat{\mathcal{V}}^\ell$ is obtained by passing \mathcal{V}^ℓ to the encoder and decoder steps.

Once $\hat{\mathcal{V}}^\ell$ is obtained, two distinct cases are considered, namely *no back-scaling (NBS)* and *back-scaling (BS)*. Let $\hat{\mathbf{p}}^\ell = (\hat{p}_Y^\ell, \hat{p}_U^\ell, \hat{p}_V^\ell) \in \hat{\mathcal{V}}^\ell$ and $\hat{\mathbf{p}} = (\hat{p}_Y, \hat{p}_U, \hat{p}_V) \in \hat{\mathcal{V}}$ be pixels which belong to $\hat{\mathcal{V}}^\ell$ and $\hat{\mathcal{V}}$ used as input of the machine vision task, respectively. The NBS consists of omitting the luma back-scaling step, thus:

$$\forall \hat{\mathbf{p}}^\ell = (\hat{p}_Y^\ell, \hat{p}_U^\ell, \hat{p}_V^\ell) \in \hat{\mathcal{V}}^\ell, \quad \hat{\mathbf{p}} = \hat{\mathbf{p}}^\ell. \quad (2)$$

As a result of (2), the luma channel of $\hat{\mathcal{V}}$ that serves as input of the machine vision task remains downscaled. As opposed

TABLE I. SUMMARY OF DATASETS, SEQUENCES, MACHINE ARCHITECTURES, AND ASSESSMENT METRICS FOR OBJECT DETECTION AND TRACKING, AS DEFINED IN THE CTC [16].

Machine vision task	Dataset	Sequence		Machine vision architecture	Assessment metric	
		Count	Resolution			
Object detection	SFU-HW-objects-v1 [20]	Class A	1	2560×1600	Faster R-CNN [24] with a ResNeXt-101 [23] backbone from detectron2 [21] library	mean average precision (mAP)
		Class B	4	1920×1080		
		Class C	4	832×480		
		Class D	4	416×240		
Object tracking	Tencent Video Dataset (TVD) [25]	7	1920×1080	joint detection and embedding (JDE) [22]	multi-object tracking accuracy (MOTA)	

to NBS, the BS case aims to revert the luma down-scaling step from (1) to obtain the pixel $\hat{\mathbf{p}}$ from $\hat{\mathbf{p}}^\ell$:

$$\forall \hat{\mathbf{p}}^\ell = (\hat{p}_Y^\ell, \hat{p}_U^\ell, \hat{p}_V^\ell) \in \hat{\mathcal{V}}^\ell, \quad (3)$$

$$\hat{\mathbf{p}} = (\hat{p}_Y, \hat{p}_U, \hat{p}_V) = \left(\left\lfloor \frac{\hat{p}_Y^\ell}{\ell} \right\rfloor, \hat{p}_U^\ell, \hat{p}_V^\ell \right).$$

As shown in (3), going through the luma back-scaling step requires ℓ to be available, so it must be passed to the decoder end as metadata, e.g., by using a dedicated *supplemental enhancement information (SEI)* message. Despite the inclusion of additional information in the bitstream, it is noteworthy that the decoding process remains standard-compliant. Additionally, it is worth to emphasize that no SEI message is passed to the decoder when the NBS case is in use or the ℓ is a predefined constant on both the encoder and decoder sides. Ultimately, the final video $\hat{\mathcal{V}}$ is fed to the machine vision task step to obtain a high-level analysis of the visual content.

IV. EXPERIMENTAL SETUP

Our experimental setups adhere to the *common test conditions (CTC)* [16] as defined by JVET. Table I lists the used datasets, including sequence counts and resolutions, machine vision architectures, and assessment metrics for the object detection and tracking tasks. Sequences from the SFU and TVD datasets with a bit depth of less than 10 bits were scaled to a bit-depth of $b = 10$ using bit shifting prior to the luma down-scaling step.

The *VVC test model (VTM)* [17] version 20.0 was used to encode and decode all sequences. The applied CTC coding configurations included *all intra (AI)*, *low delay (LD)*, and *random access (RA)*. As the sole configuration adjustment, we disabled *luma mapping with chroma scaling (LMCS)* for our proposal, but it remained enabled for the anchor as per CTC. LMCS seeks for better precision during coding by scaling pixel values to occupy the available bit depth in a uniform manner. Given that both the LMCS and our proposal affect the pixel value distribution of the input, disabling LMCS isolates the effect in our proposal. Finally, as the video $\hat{\mathcal{V}}$ is in YUV format after the luma back-scaling step, and the subsequent machine vision task accepts a video in the RGB format, a color space conversion from YUV to RGB was applied with the library *ffmpeg* [18] according to the ITU-R BT.601 standard.

The performance of the proposed luma range scaling was evaluated through an extensive set of experiments. A total of 8 luma-scaling factors of $\ell \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ were considered for both the NBS and BS cases. Each of these configurations was evaluated with the pareto *Bjontegaard Delta*

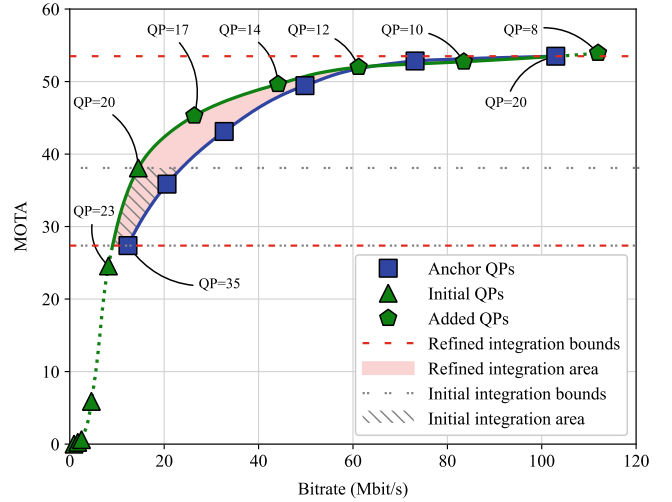


Fig. 2. The illustration of QP refinement for the sequence *TVD-01_1* under AI configuration with $\ell = 0.2$ and BS. QP values of 8, 10, 12, 14, and 17 are added on top of initial QPs $\in \{20, 23, 26, 29, 32, 35\}$ to maximize the overlapping area with the anchor.

Bitrate (BD-rate) [19] as per CTC scripts provided by JVET [16]. Note that the rate required to encode ℓ is not considered in the BS case, as incorporating a 17-byte SEI message in the bitstream for an entire video sequence is negligible.

The QPs defined in the CTC [16] are employed for the anchor. For luma down-scaled sequences, refined QPs are employed on top of the standard ones to make the computed BD-rate scores more reliable. This choice stems from the observation that scaled sequences have a much lower rate than their anchor counterparts, especially when ℓ converges towards zero ($\ell \rightarrow 0$). The utilization of refined QPs is illustrated in Fig. 2 for the *TVD-01_1* sequence under AI configuration with $\ell = 0.2$ and BS. As refined QPs are selected for each ℓ sequence-wise, explicit QPs are not reported for the sake of clarity. As a rule of thumb, QPs used to compute BD-rates are selected to maximize the overlapping with the anchor curve. Additionally, at least six QPs are employed to compute the BD-rate for each sequence and ℓ , as a greater amount of supporting points are shown to make the BD-rate scores more reliable [26].

V. EXPERIMENTAL RESULTS

Table II, Table III, and Table IV report the BD-rate results for the proposed luma range scaling technique under AI, LD, and RA configurations, respectively. The values, expressed as percentages, indicate the bitrate difference between

TABLE II.

BD-RATE SCORES FOR THE PROPOSED LUMA RANGE SCALING TECHNIQUE UNDER THE ALL INTRA (AI) CONFIGURATION

		$\ell = 0.2$		$\ell = 0.3$		$\ell = 0.4$		$\ell = 0.5$		$\ell = 0.6$		$\ell = 0.7$		$\ell = 0.8$		$\ell = 0.9$	
		NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS
Object detection	SFU Class A	-22.5%	-22.4%	-28.4%	-15.2%	-27.1%	-15.6%	-15.4%	-1.9%	-12.5%	-4.8%	-5.2%	4.4%	-6.3%	0.7%	-2.9%	-1.5%
	SFU Class B	41.7%	4.7%	-9.7%	0.0%	-16.7%	-4.5%	-17.9%	-3.2%	-15.5%	-4.8%	-10.7%	-1.4%	-10.6%	-4.1%	-2.1%	-0.9%
	SFU Class C	172.8%	-1.8%	52.3%	-3.7%	21.5%	-5.4%	10.7%	-3.7%	5.9%	-4.4%	3.4%	-4.1%	0.7%	-3.1%	-0.7%	-2.1%
	SFU Class D	109.2%	-2.8%	48.9%	-7.0%	11.2%	-6.6%	2.1%	-4.6%	-3.8%	-5.9%	-6.5%	-2.3%	-6.0%	-0.9%	-5.1%	-2.2%
	Average	97.8%	-1.7%	26.0%	-4.5%	2.8%	-6.3%	-2.8%	-3.7%	-5.1%	-5.0%	-4.7%	-2.1%	-5.4%	-2.5%	-2.7%	-1.7%
Object tracking	TVD-01_1	100.0%	-29.2%	-12.4%	-29.3%	-38.1%	-27.3%	-35.2%	-19.3%	-27.1%	-14.7%	-20.8%	-8.2%	-13.4%	-4.2%	-7.1%	2.1%
	TVD-01_2	100.0%	-31.2%	-31.9%	-28.6%	-38.6%	-20.1%	-35.5%	-14.3%	-18.8%	-6.7%	-18.6%	-8.2%	-14.9%	-4.3%	-5.4%	0.5%
	TVD-01_3	211.4%	-24.3%	-27.6%	-20.2%	-40.2%	-15.8%	-36.8%	-12.6%	-25.4%	-4.3%	-19.9%	-1.7%	-15.1%	0.2%	-3.9%	5.2%
	TVD-02_1	268.6%	17.2%	67.3%	6.8%	-10.4%	3.9%	-18.1%	-3.3%	-12.9%	-2.4%	-17.1%	-3.1%	-18.5%	2.0%	-14.4%	-5.0%
	TVD-03_1	125.8%	14.8%	-13.8%	1.5%	-29.4%	0.2%	-35.9%	1.1%	-26.4%	1.3%	-18.4%	2.3%	-10.6%	3.9%	-3.9%	2.6%
	TVD-03_2	252.1%	8.3%	27.2%	-3.2%	-23.2%	1.5%	-36.4%	-0.4%	-33.8%	-0.4%	-30.9%	-3.3%	-19.7%	4.6%	-9.9%	-0.4%
	TVD-03_3	300.8%	3.9%	53.2%	-6.2%	-7.8%	1.0%	-34.5%	-3.4%	-33.8%	-0.6%	-26.0%	0.7%	-18.2%	1.8%	-3.4%	10.2%
	Average	194.1%	-5.8%	8.8%	-11.3%	-26.8%	-8.1%	-33.2%	-7.5%	-25.5%	-4.0%	-21.7%	-3.1%	-15.8%	-0.7%	-6.8%	2.2%
	Total average	131.5%	-3.1%	20.0%	-6.9%	-7.5%	-6.9%	-13.4%	-5.0%	-12.2%	-4.6%	-10.6%	-2.4%	-9.0%	-1.9%	-4.1%	-0.4%

TABLE III.

BD-RATE SCORES FOR THE PROPOSED LUMA RANGE SCALING TECHNIQUE UNDER THE LOW DELAY (LD) CONFIGURATION

		$\ell = 0.2$		$\ell = 0.3$		$\ell = 0.4$		$\ell = 0.5$		$\ell = 0.6$		$\ell = 0.7$		$\ell = 0.8$		$\ell = 0.9$	
		NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS
Object detection	SFU Class A	-16.3%	-22.1%	-37.9%	-34.0%	-28.4%	-9.6%	-36.9%	-10.1%	-18.0%	-8.8%	-15.9%	-4.6%	-8.9%	-5.2%	-1.7%	1.7%
	SFU Class B	58.5%	-7.2%	-9.3%	-13.8%	-8.3%	-7.1%	-17.2%	-14.1%	-14.6%	-7.0%	-11.3%	-7.0%	-0.4%	2.5%	-4.3%	-1.1%
	SFU Class C	329.9%	-2.2%	-9.3%	-9.3%	-35.7%	-5.1%	-31.7%	-7.9%	-33.8%	-10.4%	-17.0%	-2.5%	-7.4%	7.3%	-19.5%	-11.6%
	SFU Class D	312.1%	-9.3%	78.1%	-12.8%	19.8%	-9.7%	7.4%	-6.7%	-10.1%	-11.9%	-2.7%	0.9%	-2.3%	1.2%	-2.1%	-0.2%
	Average	197.0%	-10.5%	36.9%	-15.0%	10.3%	-8.7%	-2.3%	-9.5%	-7.1%	-8.9%	-1.8%	-3.5%	-0.4%	-0.6%	-2.1%	-1.0%
Object tracking	TVD-01_1	100.0%	-12.4%	3.9%	-29.7%	-39.2%	-16.7%	-40.2%	-16.7%	-36.4%	-19.4%	-8.6%	3.6%	-20.5%	-2.8%	-10.0%	-1.0%
	TVD-01_2	100.0%	-18.1%	-19.9%	-12.9%	-36.7%	-17.7%	-35.7%	-20.0%	-35.5%	-27.6%	-26.2%	-23.0%	-38.3%	-24.9%	-19.3%	-11.3%
	TVD-01_3	329.9%	-2.2%	-9.3%	-9.3%	-35.7%	-5.1%	-31.7%	-7.9%	-33.8%	-10.4%	-17.0%	-2.5%	-7.4%	7.3%	-19.5%	-11.6%
	TVD-02_1	160.0%	6.0%	97.7%	27.3%	18.2%	3.4%	-0.8%	-1.8%	-2.3%	-4.8%	-9.8%	-7.8%	-11.0%	3.1%	9.0%	18.5%
	TVD-03_1	86.7%	-5.9%	-2.0%	-15.2%	-16.5%	-8.9%	-23.8%	-8.6%	-20.4%	-1.7%	-15.8%	-4.2%	-13.2%	-4.9%	-10.3%	-7.4%
	TVD-03_2	1203.5%	-10.6%	101.5%	-2.7%	1.9%	-6.3%	-32.2%	-0.9%	-39.0%	-5.2%	-30.4%	12.4%	-27.9%	-8.9%	-5.2%	8.1%
	TVD-03_3	2253.4%	3.0%	181.1%	2.7%	61.6%	16.1%	-26.6%	6.1%	-37.2%	4.0%	-29.6%	12.2%	-30.4%	-7.8%	-11.1%	9.7%
	Average	604.8%	-5.8%	50.4%	-5.7%	-6.6%	-5.0%	-27.3%	-7.1%	-29.2%	-9.3%	-19.6%	-1.4%	-21.2%	-5.6%	-9.5%	0.7%
Total average	339.7%	-8.8%	41.7%	-11.8%	4.4%	-7.4%	-11.0%	-8.7%	-14.9%	-9.0%	-8.0%	-2.7%	-7.7%	-2.3%	-4.7%	-0.4%	

TABLE IV.

BD-RATE SCORES FOR THE PROPOSED LUMA RANGE SCALING TECHNIQUE UNDER THE RANDOM ACCESS (RA) CONFIGURATION

		$\ell = 0.2$		$\ell = 0.3$		$\ell = 0.4$		$\ell = 0.5$		$\ell = 0.6$		$\ell = 0.7$		$\ell = 0.8$		$\ell = 0.9$	
		NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS	NBS	BS
Object detection	SFU Class A	135.1%	32.1%	43.9%	-4.8%	64.6%	5.2%	22.0%	11.9%	21.3%	18.8%	48.2%	27.3%	35.9%	32.8%	15.7%	11.2%
	SFU Class B	104.1%	-7.2%	-3.3%	-4.9%	-16.9%	-13.5%	-13.6%	-6.7%	-10.9%	-5.2%	-3.2%	2.5%	-9.8%	-5.0%	-11.7%	-10.9%
	SFU Class C	377.6%	-4.0%	65.3%	-8.6%	41.7%	-6.5%	22.2%	-10.2%	20.5%	-4.9%	12.7%	-3.2%	11.6%	-0.3%	4.9%	0.6%
	SFU Class D	273.2%	-4.0%	111.2%	-9.5%	32.3%	-10.6%	11.9%	-8.0%	4.2%	0.5%	-4.4%	-4.8%	-2.3%	-3.5%	3.2%	5.0%
	Average	242.7%	-2.2%	56.7%	-7.5%	22.5%	-9.0%	8.0%	-6.7%	5.9%	-1.5%	5.3%	0.4%	2.6%	-0.2%	0.1%	-0.8%
Object tracking	TVD-01_1	100.0%	-30.9%	-5.6%	-33.5%	-38.3%	-33.9%	-41.7%	-31.4%	-26.4%	-14.4%	-29.9%	-17.3%	-19.4%	-6.7%	-9.1%	-1.4%
	TVD-01_2	100.0%	-26.5%	-15.8%	-28.3%	-30.0%	-26.6%	-30.6%	-9.6%	-15.3%	-5.1%	-12.1%	2.2%	-20.9%	-10.4%	-3.4%	4.5%
	TVD-01_3	100.0%	-23.2%	-25.2%	-15.6%	-40.6%	-25.6%	-46.1%	-28.0%	-33.7%	-18.9%	-30.5%	-10.0%	-29.8%	-13.9%	-18.2%	-7.6%
	TVD-02_1	166.7%	-5.1%	48.7%	1.4%	-16.1%	15.9%	-15.2%	55.8%	-22.2%	-10.0%	2.1%	23.4%	9.1%	36.4%	18.2%	40.9%
	TVD-03_1	183.6%	3.2%	17.6%	-3.7%	9.8%	0.4%	-21.3%	-6.7%	-21.1%	-4.2%	-13.9%	0.8%	-8.0%	1.0%	-3.3%	2.9%
	TVD-03_2	1384.6%	-7.8%	92.4%	-11.9%	0.1%	-11.1%	-38.1%	-9.6%	-39.5%	-17.7%	-30.1%	-3.9%	-19.6%	1.6%	-8.3%	-1.2%
	TVD-03_3	100.0%	-16.4%	169.2%	-7.0%	41.2%	-2.7%	-23.2%	-0.5%	-38.7%	-2.5%	-33.0%	-11.5%	-16.2%	0.7%	-22.9%	-13.2%
	Average	305.0%	-15.3%	40.2%	-14.1%	-10.6%	-11.9%	-30.9%	-4.3%	-28.1%	-10.4%	-21.1%	-2.3%	-15.0%	1.2%	-6.7%	3.5%
Total average	264.5%	-6.8%	50.9%	-9.8%	10.9%	-10.0%	-5.6%	-5.9%	-6.0%	-4.6%	-3.9%	-0.5%	-3.6%	0.3%	-2.3%	0.7%	

our proposal and the anchor for equivalent task accuracy, i.e., mAP for object detection and MOTA for object tracking. Negative values indicate improved coding efficiency. The results for the SFU-HW-objects-v1 dataset are presented by class, whereas the TVD results are presented on a per-sequence basis. The averages of the SFU results are weighted to account

for the imbalance in the number of sequences in Class A over the other classes. Similarly, the total average, encompassing both SFU and TVD results, is a weighted average.

The total average results indicate that, in the general case, the highest coding gains could be achieved with a luma-scaling

factor ℓ in the range of 0.4 to 0.6. The highest average gains are -13.4% ($\ell = 0.5$ and NBS), -14.9% ($\ell = 0.6$ and NBS), and -10.0% ($\ell = 0.4$ and BS) for AI, LD, and RA configurations, respectively. However, distinct variations are observed between the datasets and back-scaling methods. For the SFU dataset, the optimal appears to be BS with $\ell = 0.4$, $\ell = 0.3$, and $\ell = 0.4$ for AI, LD, and RA configurations, respectively. The corresponding average BD-rate scores are -6.3%, -15.0%, and -9.0%. In contrast, for the TVD dataset, the optimal results are achieved using NBS with $\ell = 0.5$, $\ell = 0.6$, and $\ell = 0.5$ for AI, LD, and RA configurations, respectively. The corresponding average BD-rate scores for TVD demonstrate substantial improvements of -33.2%, -29.2%, and -30.9%.

Analyzing the object detection results on SFU at the class level reveals that Class A yields substantial BD-rate gains of up to -28.4% and -37.9% with the lowest luma-scaling factors in AI and LD configurations when NBS is applied. However, in the RA configuration, Class A predominantly results in performance degradation, with the sole exception being a luma-scaling factor of $\ell = 0.3$ with BS. Class B mostly yields optimal results, up to -17.9% BD-rate reduction, with NBS and in the luma-scaling factor range of $\ell = 0.4$ to $\ell = 0.6$. Classes C and D consistently achieve the highest gains using BS. The only exceptions occur in Class D for AI and LD configurations with luma-scaling factors ranging from $\ell = 0.7$ to $\ell = 0.9$. The optimal results for both classes, -13.8% and -12.8%, are achieved with $\ell = 0.3$ in the LD configuration.

Analysis of the object tracking results on the TVD dataset reveals a trend, where optimal average results are achieved with BS when $\ell = 0.2$ and $\ell = 0.3$, whereas NBS tends to work better from $\ell = 0.4$ upwards. The range of $\ell = 0.4$ to $\ell = 0.6$ appears to be the most optimal one for TVD sequences, although substantial coding gains are still observed at higher luma-scaling factors. For instance, the *TVD-01_2* sequence demonstrated an exceptional improvement of -38.3% with $\ell = 0.8$, in the LD configuration. In general, the object tracking results demonstrate superior coding gains, up to -46.1% in the best case.

The most closely related work by Ding *et al.* [12] truncates the least significant bit of luma values using bit shifting. Mathematically, this approach is equivalent to our method when $\ell = 0.5$, so their work can be considered a special case of our proposal. However, our results are not directly comparable because they did not refine the QPs to match bitrates with the anchor, and to the best of our knowledge, they had the LMCS enabled. Anyway, we can still use our results and compare the performance of $\ell = 0.5$ with those of other luma-scaling factors, when the LMCS is disabled. The dataset averages indicate that $\ell = 0.5$ is one of the most viable options for object tracking on the TVD dataset. Conversely, there are superior options to choose for object detection in the SFU dataset. Furthermore, when analyzing the class-wise or sequence-wise scores, alternative luma-scaling factors potentially yield better coding efficiency in most cases.

VI. CONCLUSION

In this paper, we presented a luma range scaling technique to improve coding efficiency of VVC for machine consumption. Our method down-scales the luma channel of the input video to a reduced range prior to encoding. The downscaling effectively decreases the bitrate required to encode the video and, as demonstrated by our experimental results, it reduces the accuracy of machine vision tasks less than the increment of QP during encoding.

In our experiments, an extensive range of parameters were explored to identify optimal settings. The reported BD-rate results demonstrated significant potential for coding gain. For instance, in the object detection task using the SFU dataset, BD-rate reduction of up to -37.9% was achieved with the proposed method. Similarly, the object tracking task on the TVD dataset exhibited substantial BD-rate reduction of up to -46.1%.

Future research will focus on developing adaptive methods to dynamically determine the optimal parameters for the luma range scaling technique based on content characteristics and properties of the video input. Furthermore, future efforts will integrate and evaluate the proposed method within the MPEG VCM reference software framework.

REFERENCES

- [1] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: a paradigm of collaborative compression and intelligent analytics," *IEEE Trans. Image Process.*, vol. 29, pp. 8680–8695, Aug. 2020.
- [2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [3] B. Bross *et al.*, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.
- [4] J. Chen, C. Hollmann, and S. Liu, "Optimization of encoders and receiving systems for machine analysis of coded video content (draft 3)," document JVET-AE2030-v1, Geneva, Switzerland, Jul. 2023.
- [5] L. Galteri, M. Bertini, L. Seidenari, and A. Del Bimbo, "Video compression for object detection algorithms," in *Proc. Int. Conf. Pattern Recognit.*, Beijing, China, Aug. 2018, pp. 3007–3012.
- [6] A. Zahra, M. Ghafoor, K. Munir, A. Ullah, and Z. Ul Abideen, "Application of region-based video surveillance in smart cities using deep learning," *Multimed Tools Appl.*, vol. 83, no. 5, pp. 15313–15338, Dec. 2021.
- [7] J. Xiao *et al.*, "A sensitive object-oriented approach to big surveillance data compression for social security applications in smart cities," *Softw. Pract. Exp.*, vol. 47, no. 8, pp. 1061–1080, Aug. 2017.
- [8] A. D. Bagdanov, M. Bertini, A. Del Bimbo, and L. Seidenari, "Adaptive video compression for video surveillance applications," in *Proc. IEEE Int. Symp. Multimedia*, Dana Point, CA, USA, 2011, pp. 190–197.
- [9] J. Laitinen *et al.*, "Feasibility study of multi-layer VVC coding scheme for hybrid machine-human consumption," in *Proc. IEEE Int. Conf. Multimedia Expo*, Niagara Falls, Canada, Jul. 2024.
- [10] A. Aliouat, N. Kouadria, M. Maimour, and S. Harize, "An efficient low complexity region-of-interest detection for video coding in wireless visual surveillance," in *Proc. Int. Multi-Conf. Syst., Signals & Devices*, Setif, Algeria, May 2022, pp. 1357–1362.
- [11] A. Marie, K. Desnos, L. Morin, and L. Zhang, "Video coding for machines: large-scale evaluation of deep neural networks robustness to compression artifacts for semantic segmentation," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Shanghai, China, Sep. 2022, pp. 01–06.

- [12] D. Ding, X. Zhao, and S. Liu, "Truncating bit depth in video coding for machine tasks," *document JVET-AG0178-v4*, Online, Jan. 2024.
- [13] H. Choi and I. V. Bajic, "High efficiency compression for object detection," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, Calgary, Canada, Apr. 2018, pp. 1792–1796.
- [14] Q. Cai, Z. Chen, D. O. Wu, S. Liu, and X. Li, "A novel video coding strategy in HEVC for object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4924–4937, Dec. 2021.
- [15] K. Fischer, F. Fleckenstein, C. Herglotz, and A. Kaup, "Saliency-driven versatile video coding for neural object detection," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, Toronto, Canada, Jun. 2021, pp. 1505–1509.
- [16] S. Liu and C. Hollman, "Common test conditions for optimization of encoders and receiving systems for machine analysis of coded video content," *document JVET-AF2031-v1*, Hannover, Germany, Oct. 2023.
- [17] "VVC Reference Software Version 20.0," Accessed: Aug. 2024. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-20.0.
- [18] "FFmpeg," Accessed: Aug. 2024. [Online]. Available: <https://ffmpeg.org>.
- [19] G. Bjøntegaard, "Improvements of the BD-PSNR model," *document VCEG-A111*, Berlin, Germany, Jul. 2008.
- [20] H. Choi, E. Hosseini, S. Ranjbar Alvar, R. A. Cohen, and I. V. Bajic, "A dataset of labelled objects on raw video sequences," *Data in Brief*, vol. 34, 2021.
- [21] Y. Wu, A. Kirillov, F. Masa, W.-Y. Lo, and R. Girschick, "Detectron2," 2019. Accessed: May 2024. [Online]. Available: <https://github.com/facebookresearch/detectron2>.
- [22] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vision*, Aug. 2020, pp. 107–122.
- [23] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Honolulu, Hawaii, Jul. 2017, pp. 1492–1500.
- [24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object Detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, Montréal, Canada, Dec. 2015, pp. 91–99.
- [25] W. Gao, X. Xu, M. Qin, and S. Liu, "An open dataset for video coding for machines standardization," in *Proc. IEEE Int. Conf. Image Process.*, Bordeaux, France, Oct. 2022, pp. 4008–4012.
- [26] C. Herglotz et al., "The Bjøntegaard bible why your way of comparing video codecs may be wrong," *IEEE Trans. Image Process.*, vol. 33, pp. 987–1001, Jan. 2024.