



**HAL**  
open science

# Energy Consumption and Carbon Emissions of Modern Software Video Encoders

Taieb Chachou, Wassim Hamidouche, Sid Ahmed Fezza, Ghalem Belalem

► **To cite this version:**

Taieb Chachou, Wassim Hamidouche, Sid Ahmed Fezza, Ghalem Belalem. Energy Consumption and Carbon Emissions of Modern Software Video Encoders. IEEE consumer electronics magazine, 2023, Ieee Consumer Electronics Magazine, pp.1-16. 10.1109/mce.2023.3347714 . hal-04385234

**HAL Id: hal-04385234**

**<https://hal.science/hal-04385234>**

Submitted on 15 Mar 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Energy Consumption and Carbon Emissions of Modern Software Video Encoders

Taieb Chachou  
Université Oran1

Wassim Hamidouche  
Univ. Rennes, INSA Rennes, CNRS, IETR - UMR  
6164  
Technology Innovation Institute

Sid Ahmed Fezza  
National Higher School of Telecommunications  
and ICT

Ghalem Belalem  
Université Oran1

**Abstract**—In today’s digital landscape, video streaming holds an important role in internet traffic, driven by the pervasive use of mobile devices and the surge in streaming platform popularity. In light of this, the imperative to gauge energy consumption takes center stage, paving the way for eco-conscious and sustainable video streaming solutions with a minimal Carbon Dioxide (CO<sub>2</sub>) footprint. This paper meticulously examines the energy consumption and CO<sub>2</sub> emissions of five popular open-source and fast video encoders: x264, x265, VVenC, libvpx-vp9, and SVT-AV1. These encoders are optimized software implementations of three video coding standards (AVC/H.264, HEVC/H.265, VVC/H.266) and two video formats (VP9 and AV1). To ensure a fair comparison, we also assess coding efficiency across these encoders at four distinct presets, applying three objective quality metrics. Additional factors like computing density and memory usage are considered. Our experiments employ the JVET-CTC video dataset, encompassing video sequences of diverse content and resolutions. Encoding is executed on an Intel x86 multi-core processor, while CO<sub>2</sub> emissions are computed based on the energy mix data from a server situated in France, reflecting an average emission rate of 101 g/kWh. Our findings underscore that the x264 and SVT-AV1 encoders, especially at fast and faster presets, exhibit the lowest energy consumption and CO<sub>2</sub> emissions. Notably, x264 boasts the most energy-efficient performance, yielding CO<sub>2</sub> emissions of 0.28 g, 0.91 g, 2.07 g, and 9.74 g when encoding videos using faster, fast, medium, and slower presets, respectively. Furthermore, SVT-AV1 and VVenC encoders operating at a slower preset demonstrate superior coding efficiency, albeit at the cost of higher computational complexity and CO<sub>2</sub> emissions of 60.5 g and 406 g, respectively. A salient observation from our study is that resolution and encoder presets serve as crucial parameters for curbing energy consumption and CO<sub>2</sub> emissions, albeit with an inherent trade-off in video quality. Comprehensive results from this research are publicly accessible [https://chachoutaieb.github.io/encoding\\_energy\\_co2](https://chachoutaieb.github.io/encoding_energy_co2).

■ **CLIMATE CHANGE** is one of the major environmental challenges caused by human-induced changes to the

global carbon cycle through emissions of greenhouse

*Date of publication 00 xxxx 0000; date of current version 00  
xxxx 0000*

*Digital Object Identifier 10.1109/MCE.2023.Doi Number*

July/August 2023

1

ACCEPTED MANUSCRIPT / CLEAN COPY

gases, particularly CO<sub>2</sub>. There is more CO<sub>2</sub> in the atmosphere today than there has been in the last two thousand years. The higher concentration of CO<sub>2</sub> has changed the Earth's energy balance and increased the global average surface temperatures, thereby altering long-term climatic trends.

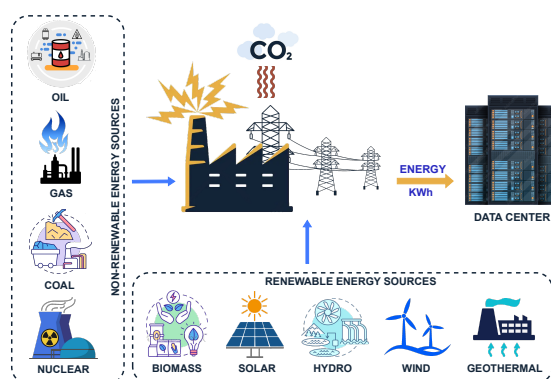
The carbon footprint of the Information and Communications Technology (ICT) sector represents a significant portion of global CO<sub>2</sub> emissions. These emissions are generated by different renewable and non-renewable energy sources used in different proportions in electricity production. For example, Figure 1 illustrates how electricity is produced from nine energy sources to power data centers. Among energy-hungry ICT applications, we can mention machine learning and video streaming. The energy consumption [1] and CO<sub>2</sub> emissions [2], [3] of machine learning algorithms have been well investigated. For instance, recently the authors of [4] have addressed the water footprint required to cool servers used for training Artificial Intelligence (AI) models (see Table 1). Moreover, with growing concerns about the energy impact of the video streaming industry, different initiatives have sprung up to understand the energy consumption related to streaming. Among them, we can cite the green of streaming [5], which is a not-for-profit members organization that focuses the streaming industry on developing joined-up engineering strategies to reduce energy waste in the delivery infrastructure.

**Table 1. Energy, carbon and water footprint of AI models.**

Reference	App.	Resource		
		Energy	CO <sub>2</sub>	Water
Strubell <i>et al.</i> (2019) [1]	NLP*	✓	✗	✗
Lottick <i>et al.</i> (2019) [2]	AI	✓	✓	✗
Lacoste <i>et al.</i> (2019) [3]	AI	✓	✓	✗
Li <i>et al.</i> (2023) [4]	AI	✗	✗	✓

(\*) NLP: natural language processing.

According to the Sandvine report [6], in the first half of 2022, bandwidth traffic was dominated mainly by video streaming, accounting for 65.93% of overall traffic. This high share of video traffic is primarily due to the success and popularity of streaming and gaming platforms such as Netflix, YouTube, and Twitch, as well as bandwidth-hungry video content at high resolution (4K/8K), high frame-rate, high dynamic range,

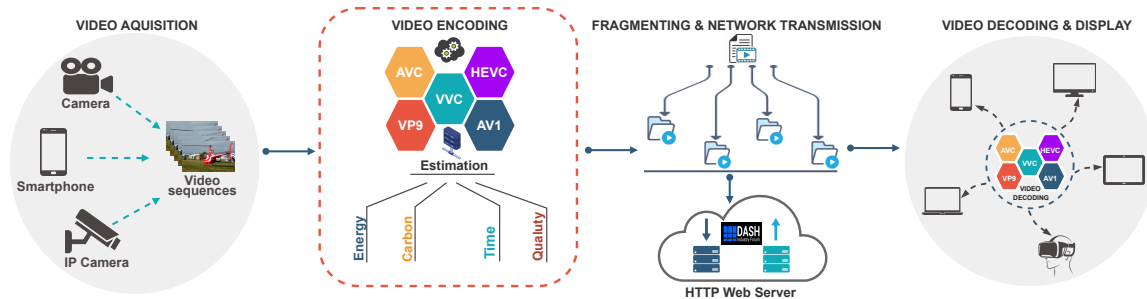


**Figure 1.** The environmental impact of energy sources used in electricity production.

and omnidirectional 360° video content. The COVID-19 pandemic has accelerated these trends even further. In Europe, YouTube traffic increased by 13%, Netflix streaming increased by 58%, and traffic from all video conferencing applications grew by 350% on the week when lockdown orders were issued [7]. Many video-streaming companies and players in this industry are striving to reduce energy consumption, as an example, we can cite the collaborative project named *dimpact*<sup>1</sup> bringing together entertainment and media companies, which aims to help the digital media industry to map and manage its carbon impacts.

Currently, video streaming is responsible for 1% of the global greenhouse gas emissions, which is the CO<sub>2</sub> emissions of a country like Spain [8]. As illustrated in Figure 2, the electronic equipment and devices ensuring end-to-end video delivery and storage are responsible for different proportions of the energy consumption associated with video streaming. Given that one video can be viewed and decoded a million times, network equipment and terminals contribute most of the total video streaming energy. International video coding standards developed over the last three decades play an essential role in reducing the size of raw videos facilitating their transmission over limited-bandwidth networks or their efficient storage in data centers [9]. The most successful and widely deployed video coding standard is by far the MPEG-4 Part 10 Advanced Video Coding (AVC)/H.264 [10], standardized in 2003 jointly by ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). These two groups have then developed new standards such as High-

<sup>1</sup><https://dimpact.org/>



**Figure 2.** End-to-end video transmission.

Efficiency Video Coding (HEVC)/H.265 [11] in 2013, and more recently, in July 2020, the latest Versatile Video Coding (VVC)/H.266 standard [12], [13]. Each new standard enhances the coding efficiency by saving up to 50% of bitrate for the same subjective video quality compared to its predecessor. In particular, the VVC/H.266 standard has demonstrated notable advancements in coding efficiency compared to its predecessor, HEVC/H.265 [14], [15]. Nevertheless, the standard improves the coding efficiency at the expense of extra complexity, especially on the encoder side. For instance, the HEVC/H.265 reference software encoder is 2 times more complex than AVC/H.264 [16], and the VVC/H.266 reference encoder is 8 times more complex than HEVC/H.265 encoder [17] under the practical Random Access (RA) coding configuration. On the other hand, the Alliance for Open Media (AOM)<sup>2</sup> was established by top tech companies to develop royalty-free and interoperable solutions for the next generation of media delivery. The AOM member companies have developed the Aomedia Video 1 (AV1) video format with coding efficiency competitive or even beyond HEVC/H.265 [15], [18].

Several works have investigated the energy consumption and optimization of video streaming at different levels of the video delivery cycle: encoding [19]–[24], transmission [25], decoding [26]–[31], and display [32], as illustrated in Figure 2 and summarized in Table 2. For instance, in 2016, MPEG released the energy-efficient media consumption standard ISO/IEC 23001-11 [33], also known as green metadata. The standard defines a set of metadata that enables controlling the energy consumption of encoding, decoding, and display for more sustainable video consumption.

However, developing an accurate energy consumption model that considers the life cycle of a video, from

**Table 2.** Energy emissions and carbon footprint estimation of video streaming applications.

Reference	Video application				Resource	
	Enc.	Dec.	Tran.	Dis.	Energy	CO2
[19]–[24]	✓	✗	✗	✗	✓	✗
[23], [27]–[30], [34]	✗	✓	✗	✗	✓	✗
[31]	✗	✓	✗	✗	✓	✓
[25]	✗	✗	✓	✗	✓	✗
[32]	✗	✗	✗	✓	✓	✗
Our	✓	✗	✗	✗	✓	✓

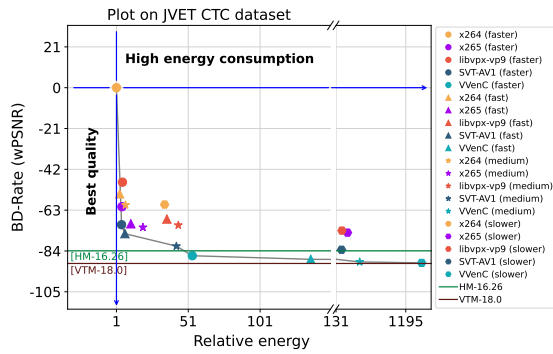
Enc.: Encoding, Dec.:Decoding, Tran.: Transmission, Dis.: Display

acquisition to display, is still challenging, as several parameters (i.e., production, maintenance, transportation of servers and devices) can have an impact on the energy footprint, requiring the availability of large-scale data from top tech streaming companies. Moreover, none of these solutions have considered all modern video encoders nor estimated their CO2 emissions. Therefore, in this paper<sup>3</sup>, we propose to focus on the energy consumption and CO2 emissions of popular open-source and fast video encoders, namely x264, x265, VVenC, libvpx-vp9, and SVT-AV1, which are based on three video coding standards (AVC/H.264, HEVC/H.265, VVC/H.266) and two video formats (VP9, and AV1). The contributions of the paper are summarized as follows:

- Assess the complexity, density, memory usage, energy consumption, and CO2 footprint of five video modern encoder implementations for encoding video sequences with different contents, resolutions, and bitrates.
- Assess the encoding trade-off between energy con-

<sup>2</sup>AOM: <https://aomedia.org/about/>

<sup>3</sup>Project page: [https://chachoutaieb.github.io/encoding\\_energy\\_co2](https://chachoutaieb.github.io/encoding_energy_co2)



**Figure 3.** BD-Rate (wPSNR) versus relative energy consumption for the five software encoders at four presets. The gray line (i.e., Pareto front) highlights configurations providing the best trade-off between compression efficiency and energy consumption.

sumption and coding efficiency based on different quality metrics, including PSNR, SSIM, and VMAF.

- Analyze the impact of video resolution and encoder preset on CO<sub>2</sub> emissions, energy consumption, and coding efficiency.
- Provide an in-depth analysis of the pros and cons of five open-source software video encoders at four different presets or speed settings.

The experimental results depicted in Figure 3 demonstrate the compelling performance of the SVT-AV1 encoder in terms of the trade-off between compression efficiency and energy consumption. VVenC and x264 encoders follow closely behind. Notably, among the top nine configurations, three were accomplished by SVT-AV1 and four by VVenC encoders, while two were achieved by the x264 encoder.

## RELATED WORK

This section briefly reviews works dealing with the measurement, consumption, and optimization of the decoder and encoder energy.

### Decoder

The first in-depth energy consumption comparative study between HEVC/H.265 and VVC/H.266 reference software decoders was conducted in [35]. The authors showed that the VVC decoder consumes 80% higher energy than the HEVC/H.265 decoder. An energy-aware VVC/H.266 encoder configuration was also proposed by disabling a set of coding tools.

This coding configuration reduces the decoder energy consumption by 17% on average at the cost of around 10% bitrate overhead. An exploration algorithm based on greedy search was proposed in [36] to select the VVC/H.266 encoding profile that minimizes the decoder-energy consumption while preserving the coding efficiency of the reference encoder that includes all VVC/H.266 tools. The derived profiles tested on the VVC/H.266 reference software encoder and decoder enable different trade-offs between decoder energy saving and bitrate loss. Křanzler *et al.* [22] analyzed the impact of VVC/H.266 encoder configurations on the decoder energy consumption using the fast Fraunhofer Versatile Video Encoder (VVenC)<sup>4</sup> and the real-time VVdeC<sup>5</sup> decoder. Two new coding profiles were derived from this study, enabling 25% and 34% decoder energy savings at the cost of 5% and 15% bitrate increases, respectively. Authors in [27] proposed two linear energy prediction models for HEVC/H.265 decoder based on features extracted from the HEVC/H.265 bitstream. The best model, tested on the reference software HEVC/H.265 decoder, achieved an energy prediction accuracy higher than 88%. More recently, Saha *et al.* [34] measured and analyzed the energy consumption of two optimized VVC/H.266 software decoders, namely OpenVVC<sup>6</sup> and VVdeC, on two resource-constraint embedded NVIDIA devices fitted by ARM processors. The two decoders showed very similar performance in decoding frame-rate and energy consumption, while OpenVVC used lower maximum memory than VVdeC.

Work in [25] tackled the energy consumption of both content production (headend) and mobile user (terminal) for video streaming with different adaptation methods on CPU and GPU. The authors identified that adaptive frame-rate is the most energy-efficient adaptive representation compared to varying spatial resolution and quality. Moreover, encoding on GPU results in additional energy saving at the headend. Authors in [30] proposed to extend the traditional encoder rate-distortion optimization to consider the decoding energy. Thus, the HEVC/H.265 encoder selects a set of coding tools that minimize the trade-off between rate distortion and decoding energy. This latter is estimated at the encoder by a bitstream feature-based model. Experimental results showed that this solution reaches

<sup>4</sup>VVenC: <https://github.com/fraunhoferhhi/vvenc>

<sup>5</sup>VVdeC: <https://github.com/fraunhoferhhi/vvdec>

<sup>6</sup>OpenVVC: <https://github.com/OpenVVC/OpenVVC>

**Table 3. Characteristics of works targeting energy consumption of video encoders.**

Ref. (Year)	Standard (Encoder)	Preset	Dataset	Hardware	Quality metric	CO2
[19] (2016)	AVC (x264), HEVC (Kvazaar, x265) VP9 (Vpxenc)	1 preset	1 HD video (Big Buck Bunny)	x86 Intel Core i7-5820K @ 3.3 GHz	PSNR, SSIM	✗
[20] (2017)	HEVC (x265)	10 presets	10 HD videos	Xeon E3-1271 @ 3.6 Ghz	PSNR	✗
[21] (2021)	AVC (x264), HEVC (x265)	9 presets	One 2s video segment	Cloud x86 and ARM	✗	✗
[22] (2022)	HEVC (x265)	9 presets	JVET-CTC, Classes: B, C, D	-	✗	✗
[23] (2022)	HEVC (x265) , VVC (VVenC) VP9 (Vpxenc), AV1 (SVT-AV1)	1 preset	JVET-CTC, Classes: B, C, D	Intel Comet Lake-S @ 3.3 GHz	PSNR, VMAF	✗
[24] (2023)	HEVC (x265)	9 presets	VCD dataset (500 UHD videos)	-	VMAF	✗
Our	AVC (x264), HEVC (x265), VVC (VVenC) VP9 (Vpxenc), AV1 (SVT-AV1)	4 presets	JVET-CTC, Classes: A1, A2, B, C, D, E, F	Xeon(R) W-2125, @ 4 GHz	PSNR, SSIM VMAF	✓

up to 30% of decoding energy saving. However, depending on the video sequence, a significant bitrate overhead of 20% to 50% is introduced.

### Encoder

The energy consumption and the coding efficiency performance of four fast video encoders, including x264, Kvazaar, x265, and libvpx-vp9, were analyzed in [19]. The experimental results showed that in contrast to the x265 encoder, the x264 encoder is the most efficient in energy consumption but provides the lowest compression efficiency compared to other encoders, while the libvpx-vp9 encoder achieves the best trade-off between energy consumption and coding efficiency. Authors in [20] conducted an in-depth energy consumption analysis of the x265 encoder considering ten presets targeting a multicore x86 processor. The performance results in energy consumption and coding efficiency are provided to guide the choice of the best encoding profile according to the target application specifications and requirements in terms of speed, latency, energy, and coding efficiency. Authors in [21] investigated the energy consumption, cost, and coding efficiency of video encoding on the new *Graviton2* ARM instances introduced in the Amazon EC2 services. A set of coding recommendations is provided to reduce the encoding cost while achieving the highest coding efficiency with the lowest energy consumption. The authors recommend using ARM-based and x86-based instances for respectively x264 and x265 encoders to reach the cheapest cost encoding. The low encoding cost is achieved thanks to the intensive use of low-level optimizations through Single Instruction Multiple Data (SIMD) by the x265 encoder for only x86 platforms. Authors in [23] performed an analysis and comparison of the energy consumption among four state-of-the-art codecs, namely VP9, x265, VVenC/VVdeC, and SVT-AV1. The experimental find-

ings demonstrated that SVT-AV1 exhibited the best trade-off between coding efficiency and energy consumption when compared to the other codecs. However, it is worth noting that the comparison between the encoders was conducted solely based on a single preset, which varied across different encoders. In a recent study, the authors of [24] proposed to include the energy consumption of the encoding process in the optimization of bitrate ladders. To achieve this, they conducted a comprehensive analysis of energy consumption and compression efficiency of different x265 presets for 500 video sequences. Then, based on their findings, a per-title model was developed to optimize the trade-off between compression efficiency and energy consumption. The authors mentioned that considering energy consumption in optimizing bitrate ladders can provide more efficient and sustainable video streaming. In a separate work, the authors of [22] proposed a simple linear model for predicting the energy consumption of the x265 encoder at different presets. Their model relied on pre-encoding with the x265 encoder using the *ultrafast* preset, which enabled high prediction accuracy across various platforms. The energy prediction model reached an accuracy of 88% on average over all presets. As a result, this model can be used to optimize and deploy energy-efficient video encoding algorithms.

This paper investigates, in addition to the energy consumption, the CO2 emissions of five leading open-source software video encoders at four presets. The lack of data on the CO2 emissions of video encoders in the literature motivates us to fill this gap with an in-depth study and analysis of their CO2 emissions. Further, we explore and compare the complexity, coding efficiency, density (i.e., parallelism efficiency), and memory usage of encoders, highlighting the pros and cons of each encoder. Table 3 summarizes the main characteristics of described works tackling energy con-

sumption of fast software video encoders.

## SYSTEM DESCRIPTION

In a digital video transmission system, the captured video sequence is processed by several stages, such as encoding/transcoding, network adaptation, transmission, and decoding, before being viewed by the end user, as illustrated in Figure 2. This section first gives the encoding data generation and the configuration each encoder uses for measuring energy consumption. Second, the measurement procedure adopted is presented for both energy consumption and CO<sub>2</sub> emissions.

### Encoding data generation

We conducted video encoding on the dataset defined by the Joint Video Exploration Team Common Test Conditions (JVET-CTC) [37] using the five considered encoders at four different bitrates, applying Constant QP (CQP) mode for VVenC and Constant Rate Factor (CRF) mode for the remaining encoders. The encoding process was carried out using four different presets or speed settings: *faster*, *fast*, *medium* and *slower*. Typically, in modern encoders, a set of presets is provided, which allows easier adjustment of certain encoding features [38]. For instance, a slower encoding preset takes more time to encode a video while providing higher compression efficiency. On the other hand, a faster encoding preset sacrifices some of the compression efficiency for faster encoding. This can be useful when speed is more important than quality, such as live streaming or real-time communication applications. The encoding process was performed on a Central Processing Unit (CPU) instance using the open-source tool FFmpeg for four encoders, including x264, x265, libvpx-vp9, and Scalable Video Technology for AV1 (SVT-AV1), while the VVenC software encoder implementation is used for the VVC/H.266 standard. Table 4 gives the command line used for each encoder, and below we describe the encoders and their coding configuration used for our experiments.

**1) Advanced Video Coding (AVC)/H.264:** For assessing the AVC/H.264 standard, we have chosen to use the x264 encoder implementation. x264 is an open-source software library designed for real-time applications. Its Application Programming Interface (API) allows configuring specific video characteristics such as the frame-rate, bitrate, bit-depth, resolution,

and the maximum number of frames to be encoded, etc. The API also provides a flexible trade-off between compression performance and computational complexity by using different presets. In this work, we encoded the videos with four different presets, namely *veryslow*, *medium*, *veryfast* and *ultrafast*.

### 2) High-Efficiency Video Coding (HEVC)/H.265:

For the HEVC/H.265 standard, we opted for the x265 implementation developed by the MulticoreWare company. x265 encoder is one of the most popular encoders for HEVC/H.265, and it supports most of the features of x264, including all bitrate control modes, frame-rate change, single-pass or multi-pass and video buffering bitrate control. As for x264, we performed the encoding with the following presets: *veryslow*, *medium*, *veryfast* and *ultrafast*.

### 3) VP9:

To test the coding performance of VP9, we used the latest version of the VP9 encoder developed by Google named libvpx-vp9. This latter is available as a codec with Fast Forward Moving Picture Experts Group (FFmpeg) tool, enabling fast VP9 encoder implementation. Furthermore, similar to x265 and x264, libvpx-vp9 supports a multipass mode, which results in improved rate-distortion performance. In addition, libvpx-vp9 provides a flexible trade-off between coding efficiency and computational complexity using speed encoding rate control. In this study, we considered four-speed encoding rates, 0, 3, 6, and 8, which are similar to the *veryslow*, *medium*, *veryfast*, and *ultrafast* presets, in x264 and x265 encoders, respectively.

### 4) Aomedia Video 1 (AV1):

The most popular fast AV1 encoder implementation is SVT-AV1, which has been integrated into the FFmpeg tool. The SVT-AV1 encoder was adopted by the AOM Software Implementation Working Group (SIWG) in August 2020. SVT-AV1 is a CPU-based multi-threaded video encoder, and it provides many encoding features, such as multiple encoding presets, bitrate control modes, video scaling, etc. In our experiments, we specifically configured the encoding preset for different performance levels. We assigned the values of 3, 6, 10, and 12 to represent the *slower*, *medium*, *fast*, and *faster* encoding presets, respectively. For this encoder, preset 0 also provides a high encoding efficiency. Still, the encoding time is much higher than other encoders, such as libvpx-vp9 and x265,

**Table 4. Command lines used to run the five video encoders considered in the experiments.**

Standard	SW encoder	Version	Command line
AVC/H.264	x264	0.164.3101	<code>ffmpeg -y -f rawvideo -pix_fmt {yuv420p, yuv420p10le} -s:v {WXH} -r {fps} -i input.yuv -c:v libx264 -pix_fmt {yuv420p, yuv420p10le} -preset {veryslow, medium, veryfast, ultrafast} -frames:v {#frames} -crf {QP} -x264-params "keyint={GOP_size}:min-keyint={GOP_size}" output.mp4</code>
HEVC/H.265	x265	3.5-2	<code>ffmpeg -y -f rawvideo -pix_fmt {yuv420p, yuv420p10le} -s:v {WXH} -r {fps} -i input.yuv -c:v libx265 -pix_fmt {yuv420p, yuv420p10le} -preset {veryslow, medium, veryfast, ultrafast} -frames:v {#frames} -crf {QP} -x265-params "keyint={GOP_size}:min-keyint={GOP_size}" output.mp4</code>
VVC/H.266	VVenC	1.9.0	<code>vvencFFapp -c randomaccess_{slow, medium, fast, faster}.cfg -c sequence.cfg</code>
VP9	libvpx-vp9	1.12.0	<code>1) ffmpeg -y -f rawvideo -pix_fmt {yuv420p, yuv420p10le} -s:v {WXH} -r {fps} -i input.yuv -frames:v {#frames} -c:v libvpx-vp9 -pix_fmt {yuv420p, yuv420p10le} -speed {0, 3, 6} -b:v 0 -crf {QP} -g {GOP_size} output.mp4 2) ffmpeg -y -f rawvideo -pix_fmt {yuv420p, yuv420p10le} -s:v {WXH} -r {fps} -i input.yuv -speed 8 -frames:v {#frames} -c:v libvpx-vp9 -pix_fmt {yuv420p, yuv420p10le} -quality realtime -b:v 0 -crf {QP} -g {GOP_size} output.mp4</code>
AV1	SVT-AV1	1.7.0	<code>ffmpeg -y -f rawvideo -pix_fmt {yuv420p, yuv420p10le} -s:v {WXH} -r {fps} -i input.yuv -frames:v {#frames} -c:v libsvtav1 -pix_fmt {yuv420p, yuv420p10le} -preset {3, 6, 10, 12} -crf {QP} -g {GOP_size} output.mp4</code>

with the *veryslow* preset.

**5) Versatile Video Coding (VVC)/H.266:** For the VVC/H.266 standard, the encoding was performed using the VVenC. This latter is a fast and efficient VVC/H.266 software encoder implementation developed by Fraunhofer Heinrich Hertz Institute (HHI). It provides a range of practical features, including perceptual optimization based on the XPSNR metric, Variable Bit-Rate (VBR) control mode along with five predefined presets. The encoder further supports multi-threading, single- and two-pass rate control modes, and task-based parallelization. For VVenC, we have set the encoding preset to *slow*, *medium*, *fast*, and *faster*. Like SVT-AV1, the encoding time of *slower* preset of VVenC is very high and is not comparable to the encoding time of the other encoders. To this end, we selected the *slow* preset to ensure high coding efficiency at a reasonable encoding time.

### Energy and CO2 footprint measurement

**Energy and power.** In physics, energy is the ability of a physical system to do a task. The computation of the amount of electrical energy transferred to an appliance depends on its power consumption and the duration required to process the task. The electrical energy can be computed by multiplying the power by time and is expressed in kilowatt-hour (kWh), referring to the energy consumed at a rate of one kilowatt for one hour.

**Energy measure and power computation.** In our work, we employed the CodeCarbon<sup>7</sup> [3] tool to measure the energy consumed during the encoding process. This tool utilizes the Running Average Power Limit (RAPL) interface integrated into Intel processors. RAPL is an interface to report accumulated

power and energy consumption of various System-on-Chip (SoC) power domains. RAPL is not an analog power meter but uses a software power model. This software power model measures energy usage relying on a set of hardware counters and I/O models. In the Linux operating system, there are four ways to access the RAPL interface data as described in [40]. The CodeCarbon tool exploits the files under `/sys/class/powercap/intel-rapl/intel-rapl:0` using the powercap interface. The RAPL file contains the energy consumption value from the register status, expressed in microjoules (mj). The energy consumption value is updated roughly every millisecond (ms), which is very suitable for the encoding process that takes more than 2 seconds to encode a video sequence of 10 seconds. Over time, the value of energy is increased to the point of overflow and then resets. The CodeCarbon tool measures energy consumption over a specified time interval by capturing two readings with a time delay and then calculates the wattage based on the difference in energy consumption and the time delay. In our work, we perform multiple measurements until the confidence interval is small to ensure statistical correctness.

**CO2 footprint computation.** To convert the energy consumption determined via the CodeCarbon tool into CO2 footprint, it is first necessary to determine the geographic location of computation servers, this can be done via their IP address using the GeoJS<sup>8</sup> API or manually by providing the country of the server location. Then, the energy mix data for that location are used. It is important to note that this conversion relates precisely to the electricity consumption. In our work, the video encoding server is located in France. The energy mix data are obtained from the electricity maps data [39], which contain individual country

<sup>7</sup><https://github.com/mlco2/codecarbon> (version 2.3.1)

<sup>8</sup><https://www.geojs.io/>



**Table 5. Electricity production ratio (EPR) of various energy sources linked to CO2 emissions intensity presented by country in 2022 [39]. EPR refers to the share of each energy source in the total electricity production. CO2 emissions intensity (CEI) is the amount of CO2 emitted per unit of electricity generated.**

Energy source		Oil	Coal	Gas	Biomass	Geothermal	Solar	Hydro	Wind	Nuclear	CO2eq (g/kWh)
France	CEI (g/kWh)	1014	954	625	230	38	30	11	13	5	101
	EPR	0.0038	0.0063	0.0874	0.135	0	0.037	0.0657	0.0764	0.575	
USA (Texas)	CEI (g/kWh)	1315	1081	513	15	109	26	24	11	12	408
	EPR	0	0.1669	0.4239	0	0	0.0549	0.0007	0.249	0.0971	
Australia (Sydney)	CEI (g/kWh)	650	820	490	230	38	45	24	11	12	586
	EPR	0	0.6819	0.0365	0	0	0.1394	0.0524	0.0897	0	

**Note:** These data are subject to change over time and the obtained results can change accordingly.

CO2 emissions and the percentage of electricity produced from nine different energy sources: oil, coal, gas, biomass, geothermal, solar, hydro, wind, and nuclear (see Figure 1). The carbon emissions per unit of power generated by energy sources are higher for the primary fossil fuels, including coal, oil, and natural gas, as presented in Table 5. This table also shows a high variation in CO2 emissions for different geographical regions.

To calculate the equivalent CO2 (CO2eq) emissions for 1 kWh of energy consumed in a country, we used Equation (1).

$$CO2eq = \sum_{s \in \mathbb{S}} EPR_s CEI_s, \quad (1)$$

where  $EPR_s$  and  $CEI_s$  are the electricity production ratio and the CO2 emissions intensity of a source  $s$ , respectively.  $\mathbb{S}$  is the set of CO2 emissions' sources.

The CO2eq is the weighted average CO2 emissions of the considered energy sources for 1 kWh of energy. For example, the CO2eq emissions for 1 kWh of energy consumed by servers located in France are equal to 101 g/kWh<sup>9</sup>. In contrast, servers in Australia (Sydney) emit 586 g/kWh, which is much more than the CO2 emissions in France.

Finally, once the CO2eq is computed by Equation (1) for a particular country following data in Table 5, the carbon footprint (g) of an encoder is simply obtained by multiplying the encoding energy (kWh) by the CO2eq (g/kWh).

<sup>9</sup>101 = 0.0038×1014 + 0.0063×954 + 0.0874×625 + 0.135×230 + 0×38 + 0.037×30 + 0.0657×11 + 0.0764×13 + 0.575×5.

## EXPERIMENTAL RESULTS

### Experimental setup

The CO2 emissions of video encoding depend mainly on the encoding time (i.e., video sequences and encoder configuration), the electric grid utilized, the energy consumption of hardware, and the energy mix of the selected country. All these details are provided in this section.

**Video sequences.** In our experiments, we selected video sequences from the JVET-CTC dataset that covers a wide range of typical video content, consisting of 26 video sequences of different resolutions, frame-rates, and bit-depths. The video sequences are grouped by resolution classes (from class A1 to class F), and a large number of sequences have high spatial resolution (3840×2160) and high frame-rate (60 fps). The JVET-CTC video sequences were encoded at four bitrates by applying CQP mode for VVenC and CRF mode for the remaining encoders, resulting in a total of 2080 encoded videos (26 videos, 4 bitrates, 5 encoders, and 4 presets).

### Used hardware and energy measurements.

The encoding process was performed on a highly efficient workstation type, specifically Intel(R) Xeon(R) W-2125 with a 4 Core CPU running at 4.00 GHz with four DDR4 RAM modules, each one with a size of 16 GB on Ubuntu 20.04.5 LTS Operating System (OS). The processor supports a wide range of SIMD, including SSE4.2, AVX, AVX2, and AVX-512. All five encoders rely heavily on assembly and intrinsic to take advantage of low-level CPU and SIMD instructions (SSE, AVX, AVX-512), which help speed up the encoding process. We used the default configuration of x264, x265, libvpx-vp9, and

SVT-AV1 encoders provided with the FFmpeg library version 6.0. The proper SSE/AVX optimizations were enabled at run time according to the instructions supported by the CPU. For VVenC, we also compiled the encoder with the default configuration provided by the project GitHub through CMake.

During the encoding process, we measured the energy consumption and the equivalence CO<sub>2</sub> emissions for servers located in France. In addition, we recorded the encoding time (run time) and percentage of CPU and memory usage to analyze the computing density and complexity of the evaluated encoders.

**Encoder settings.** We considered four different presets, namely *slower*, *medium*, *fast*, and *faster*. These presets corresponded to the following settings: {*veryslow*, *medium*, *veryfast*, and *ultrafast*} for x264 and x265, encoding speeds of 0, 3, 6, and 8 for libvpx-vp9, speed settings of 3, 6, 10, and 12 for SVT-AV1, and {*slow*, *medium*, *fast*, and *faster*} for VVenC, respectively. Thus, for ease of description, in the rest of this paper, we used the terms *slower*, *medium*, *fast*, and *faster* to refer to these different encoding presets. The JVET-CTC dataset [37] underwent encoding at four different bitrates, applying CQP mode for VVenC and CRF mode for the remaining encoders. To ensure a comprehensive evaluation, we have meticulously selected four specific Quantization Parameter (QP) values for each encoder. Specifically, for x264, x265, and VVenC, we chose the QP values of JVET-CTC, namely {22, 27, 32, 37}. Similarly, for the libvpx-vp9 and SVT-AV1 codecs, we used {33, 42, 51, 59}, employing linear interpolation and adjusting the JVET-CTC QPs to maintain a consistent range of quality and bitrate. We also took great care to establish a constant Group of Pictures (GOP) size, which we consistently set at 32 frames for all the encoders. As per the JVET-CTC, the intra-period is configured to be 32 or 64 frames, depending on the video frame-rate. In addition to our careful selections of QP, GOP size, and intra-period, we chose to base our comparisons on the default GOP configurations of these encoders at four different presets. Moreover, it's worth noting that the complexity of the GOP configuration impacts not only energy consumption and the CO<sub>2</sub> footprint but also the encoder coding efficiency. By considering these metrics, we can gain valuable insights into the overall performance and environmental impact of encoder implementations.

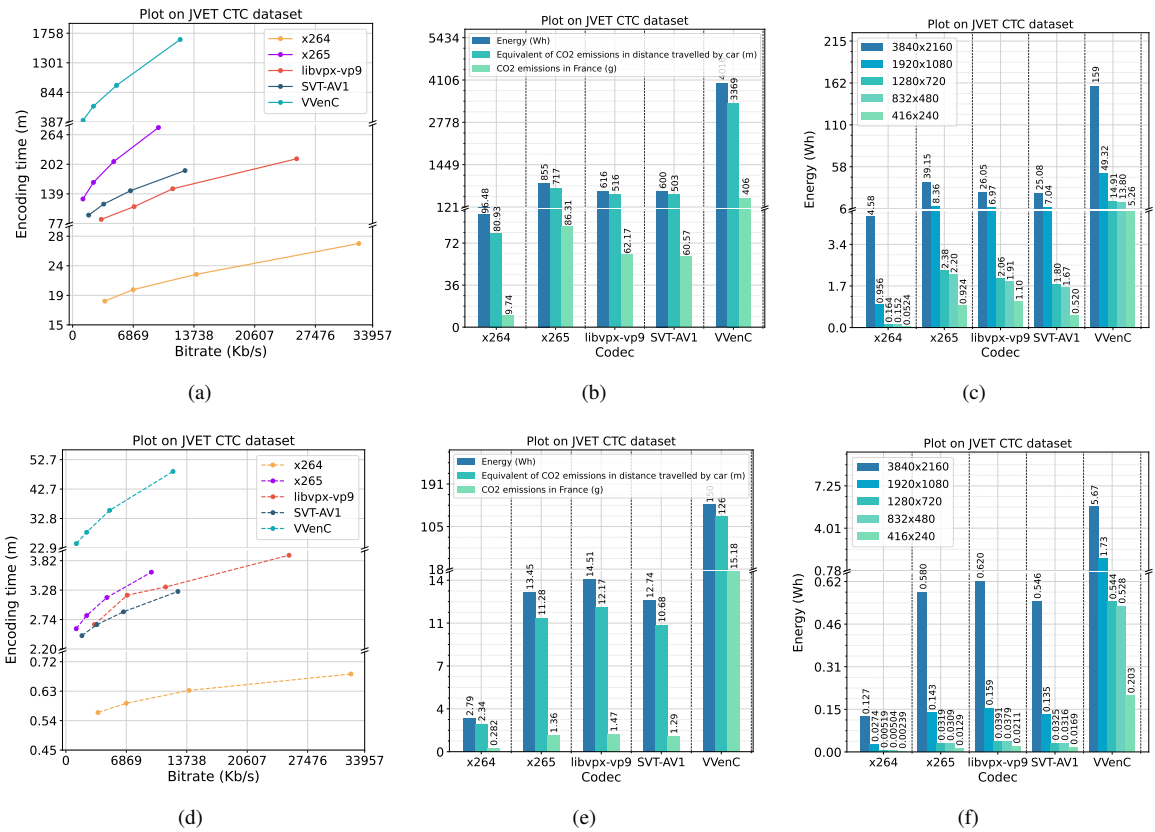
Table 4 provides an overview of the command lines utilized to execute the encoding process for the five software encoders, along with the corresponding codec versions. Finally, we provide a project's GitHub<sup>10</sup>, which contains the necessary codes allowing the reproduction of the results of this study.

**Video quality and coding efficiency.** We considered three objective full-reference quality metrics to assess the quality of the encoded videos: Peak Signal to Noise Ratio (PSNR) is a common metric that relies on the Mean Squared Error (MSE) computed over the frame pixels. Structural Similarity Index Measure (SSIM) [41] that assesses the structural similarity between the original and the encoded frame. The third considered metric is the Video Multi-method Assessment Fusion (VMAF) [42] metric, which combines human vision modeling with machine learning. To evaluate the average bitrate saving of different encoders against x264 (faster) encoder, we measured the Bjøntegaard Delta Rate (BD-Rate) [43]. This latter is commonly used for calculating the average bitrate difference in % over the whole range of rates. In our work, the x264 (faster) encoder was considered as the anchor, and the BD-Rate was computed for the three quality metrics, i.e., BD-Rate (PSNR), BD-Rate (SSIM), and BD-Rate (VMAF). For the PSNR calculation, we employed a weighted PSNR (wPSNR) metric, wherein weights of 6, 1, and 1 were assigned to the luminance and the two chrominance components, respectively.

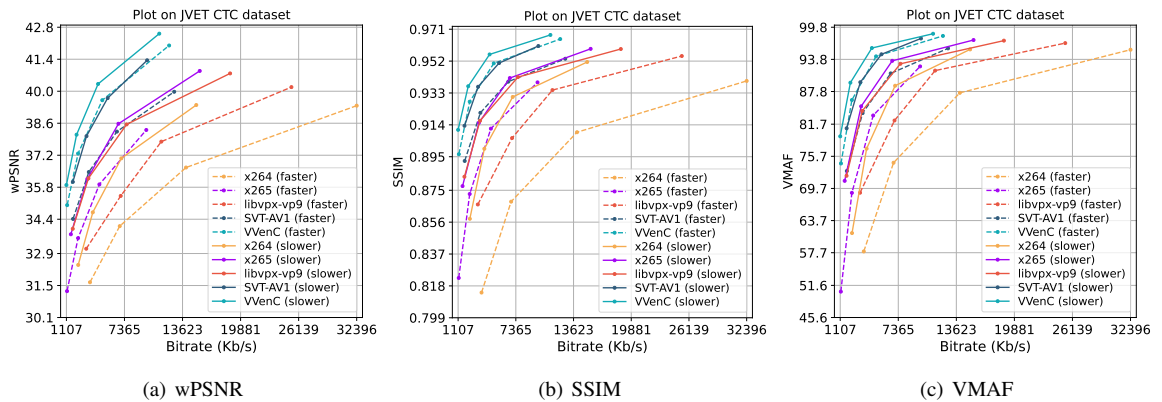
## Results and analysis

1) *Encoding complexity and coding efficiency:* To analyze the trade-off between encoding complexity and coding efficiency, we plot in Figures 4(a) and 4(d) the total encoding time for two presets, slower and faster, respectively. In addition, in Figure 5, we plot the video quality produced by the five encoders versus the bitrate. The quality is calculated using the three metrics averaged over the JVET-CTC dataset. From Figures 4(a) and 4(d), we can see that the five encoders behave similarly across all bitrates for both presets. They perform faster encoding and low quality at a low bitrate. Particularly, the x264 encoder is faster than other encoders. Nevertheless, in terms of coding efficiency, x264 (faster) achieves the lowest quality, and the VVenC (slower) encoder outperforms other

<sup>10</sup>[https://github.com/chachoutaieb/encoding\\_energy\\_co2](https://github.com/chachoutaieb/encoding_energy_co2)



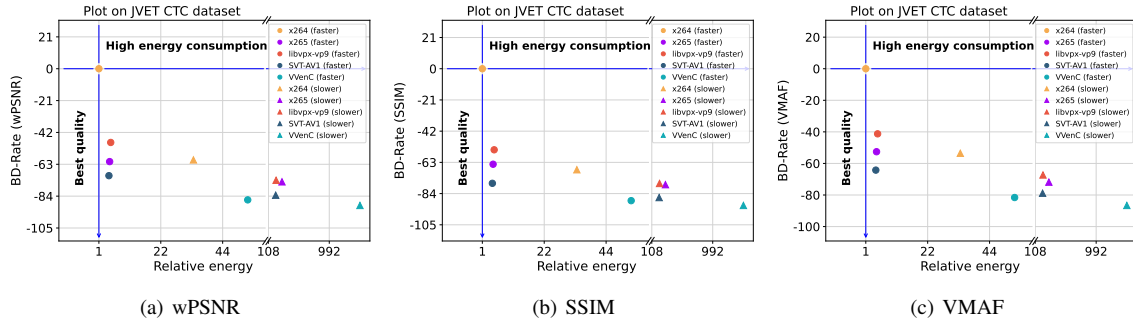
**Figure 4.** Encoding performance for slower and faster encoding presets on the first and second rows, respectively. (a) and (d) The total encoding time (run time) curve of the five encoders at different bitrates. (b) and (e) The total energy consumption, CO2 emissions in France of encoding JVET-CTC dataset, and the equivalent of the CO2 emissions in terms of distance traveled by car for the five considered encoders. (c) and (f) Total energy consumption for encoding a video of 10 seconds at different resolutions. The break (/) indicates a significant empty gap between curves.



**Figure 5.** The average quality in wPSNR (dB), SSIM and VMAF at different bitrate for the five considered encoders.

encoders on the JVET-CTC dataset. Nevertheless, the observed improvement in coding efficiency of VVenC

(slower) comes at the cost of higher encoding time. The increase in encoding time is mainly due to the new



**Figure 6.** BD-Rate versus the relative total energy consumption for the five software video encoders.

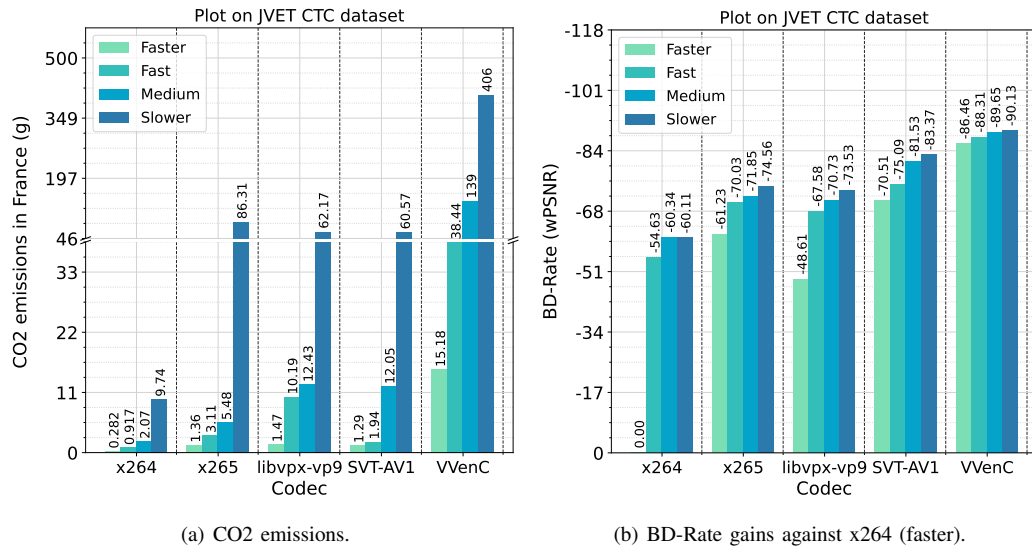
complex tools and their high search space explored during encoding to choose the most efficient coding configuration and coding tools. In addition, x264 encoder is well optimized for the x86 platform, along with efficient multi-threading parallelism. We also notice that for a faster preset, SVT-AV1 (faster) gives the best trade-off between coding efficiency and encoding complexity. Particularly, the SVT-AV1 (faster) encoder significantly reduces the encoding time while still maintaining the highest coding efficiency compared to libvpx-vp9 (faster), x265 (faster), and x264 (slower and faster).

2) *Energy consumption and CO2 emissions:* Figures 4(b) and 4(e) give the total encoding energy consumption, CO2 emissions, and the equivalent of CO2 emissions in terms of distance traveled by car for the five video encoders at slower and faster encoder settings, respectively. The total energy and CO2 emissions are measured for encoding the whole JVET-CTC dataset with four fixed QPs. To understand the CO2 emissions in the physical world, we converted the CO2 emissions to the equivalent number of meters traveled by car. In 2018, the average CO2 emissions of a passenger car registered in the European Union (EU) were about 120.4 grams of CO2 per kilometer [44]. Furthermore, we found that the energy consumption and CO2 emissions differ from one encoder to another, increasing significantly for the newborn standards. Compared to other encoders, the VVenC encoder results in significantly higher energy consumption and CO2 emissions in both slower and faster encoding presets. On the whole JVET-CTC dataset, VVenC emits an amount of CO2 equal to 406 g and 15.18 g when using slower and faster presets, respectively. These CO2 emissions are very high compared to the amount of CO2 emitted by the closest encoder in terms of encoding complexity, i.e., x265 and libvpx-vp9

when using slower and faster presets, respectively. We also notice that on both presets, SVT-AV1 minimizes energy consumption and CO2 emissions compared to x265 and libvpx-vp9. However, it is still not as efficient as the x264 encoder, which achieves the lowest energy consumption and CO2 emissions compared to other encoders, regardless of the encoder preset used. Overall, according to these results, the most suitable encoder for low energy and CO2 emissions is x264, which minimizes CO2 emissions on the whole JVET-CTC dataset at 9.74 g when using a slower preset, accounting for approximately 11.28%, 15.66%, 16.08% and 2.39% of the emissions of x265, libvpx-vp9, SVT-AV1 and VVenC encoders, respectively.

3) *Impact of the video resolution:* The total energy usage is compared across five different video resolutions to analyze the impact of video resolution on encoder energy consumption. This analysis is presented using two different presets, faster and slower, as illustrated in Figures 4(c) and 4(f), respectively. The total energy usage is measured for encoding a video of 10 seconds. The figures demonstrate that all encoders exhibit similar behavior when the video resolution increases. More specifically, the energy consumption of all encoders increases with higher video resolutions because the encoder processes more data per frame. Furthermore, it is apparent from the figures that the energy consumption of the VVenC encoder rises markedly at  $3840 \times 2160$  resolution. For the slower preset, the VVenC encoder's energy consumption at this resolution is 159 Wh. This value represents approximately 3.2, 10.6, 11.5, and 30.2 times the energy required by this encoder to encode  $1920 \times 1080$ ,  $1280 \times 720$ ,  $832 \times 480$ , and  $416 \times 240$  resolutions, respectively.

4) *Trade-off between energy consumption and coding efficiency:* To illustrate the relationship between energy consumption and coding efficiency, we provide



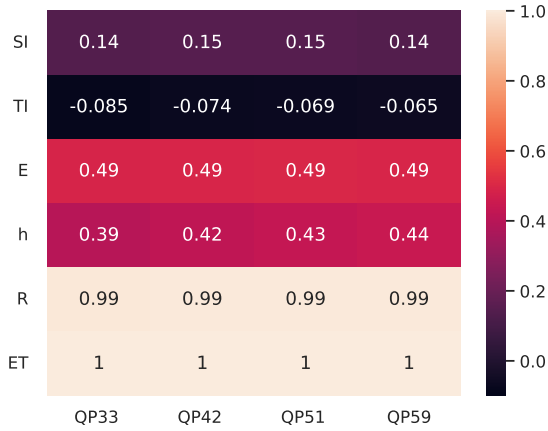
**Figure 7.** Impact of video encoding preset on CO2 emissions and coding efficiency. (a) Total CO2 emissions for encoding JVET-CTC dataset at four different presets, (b) BD-Rate (wPSNR) performance (%) of the five encoders against x264 (faster) at four different presets.

in Figure 6 the BD-Rate versus the relative energy consumption. Energy consumption and BD-Rate for the three quality metrics are presented for each encoder, using two different presets (faster and slower), compared to the anchor x264 (faster) encoder. From these figures, we can observe that compared to x264 (faster), VVenC (slower) reached the best compression performance in terms of wPSNR, SSIM, and VMAF, but this codec suffers from increased computational complexity and hence energy consumption compared to other encoders. We can also notice that in terms of energy consumption, the encoders consume low energy when using a faster preset. Among all the encoders, the SVT-AV1 (faster) encoder is the most energy-efficient relative to the x264 (faster) encoder, achieving a compelling trade-off between coding efficiency and energy consumption. On the other hand, regarding coding efficiency, the encoders using the faster preset achieve an inferior coding efficiency compared to their slower preset. The results also illustrate that when using a faster encoding preset, VVenC (faster) exhibits superior coding efficiency compared to the x264, x265, libvpx-vp9, and SVT-AV1 encoders using a slower preset, while SVT-AV1 (faster) outperforms only the x264 (slower) encoder.

5) *Impact of video encoding preset on CO2 emissions and coding efficiency:* In this evaluation, we aim to compare the total CO2 emissions across the

four different video encoding presets, i.e., slower, medium, fast, and faster. The total CO2 emissions are measured for encoding the whole JVET-CTC dataset with four fixed QPs. Additionally, the analysis takes into consideration the coding efficiency according to the encoding presets. By examining these factors, we can gain valuable insights into the trade-offs between environmental sustainability, encoding speed, and coding efficiency. From Figure 7(a), it is evident that a notable increase in CO2 emissions occurs as the encoding preset shifts from faster to slower preset. The disparity in emissions becomes more pronounced with each successive decrease in preset speed. For example, in the case of the x265 encoder, the CO2 emissions when using a faster preset amount to 1.36 g, which represents approximately, 43.7%, 24.8%, and 1.6% of the CO2 emissions when using fast, medium, and slower presets, respectively. Furthermore, it should be noted that the SVT-AV1 encoder demonstrates effective CO2 emission reduction when utilizing the fast and faster presets. In fact, SVT-AV1 outperforms other codecs in terms of minimizing CO2 emissions, second only to the x264 codec, which consistently yields lower emissions on different presets.

In terms of coding efficiency, the analysis presented in Figure 7(b) reveals a consistent trend among all encoders as the preset speed decreases. Notably, the results indicate that regardless of the encoding preset,



**Figure 8.** Pearson Correlation Coefficient (PCC) of the video complexity features (Spatial Information (SI), Temporal Information (TI), Spatial Energy (E), Temporal Energy (h)), Resolution (R), and Encoding Time (ET) with energy consumption at different QPs for the SVT-AV1 encoder with the faster preset.

VVenC provides the best wPSNR quality and BD-rate gain compared to other encoders. These results highlight the varying strengths of different encoders across different preset speeds.

6) *Impact of content complexity on energy consumption:* To investigate the impact of content complexity on energy consumption, we present in Figure 8 the Pearson Correlation Coefficient (PCC) of video complexity features, namely Spatial Information (SI) [45], Temporal Information (TI) [45], Spatial Energy (E) [46], Temporal Energy (h) [46], as well as Resolution (R), and Encoding Time (ER) with the encoding energy consumption of the SVT-AV1 (faster) encoder at different QPs. From this figure, we can observe a low correlation between spatial and temporal complexity features (SI, E, TI, h) and encoding energy consumption. In contrast, a strong correlation emerges with resolution, revealing that encoding energy consumption is highly linked to video resolution. This observation suggests that video resolution stands out as the predominant feature influencing encoding energy. To support these findings, we further explore the standard deviation of encoding energy across different resolutions, as presented in Table 6. Notably, this table reveals a small standard deviation in energy consumption for encoding videos at different resolutions, providing empirical evidence that variations in content complexity among videos of the same resolution have a negligible impact on encoding energy consumption.

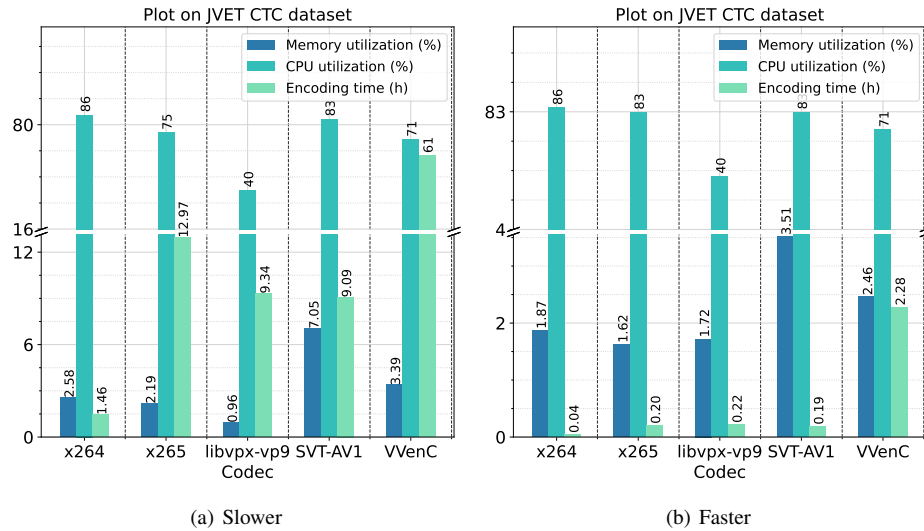
**Table 6.** Standard deviation of encoding energy at different resolutions for the SVT-AV1 encoder with the faster preset.

Resolution	#videos	Energy (Wh)	Standard deviation
416×240	4	0.0169	0.0023
832×480	5	0.0316	0.0030
1280×720	5	0.0325	0.0042
1920×1080	6	0.135	0.0079
3840×2160	6	0.546	0.042

7) *CPU and memory usage:* Encoding time performance and computing parallelism (density) are equally crucial for video encoding. In this evaluation, we provide a comparison in terms of computing density by measuring the percentage of CPU utilization for video encoding. The high CPU utilization is an indicator of the workload on the CPU. We also compare the encoders in terms of memory usage. The comparison is performed using two different encoding presets, i.e., slower and faster, as illustrated in Figures 9(a) and 9(b), respectively. From Figure 9(a), we found that x264 achieves the highest computing density when encoding video with a slower preset. These results support the fast encoding performance of the encoder, resulting in the lowest total encoding time of 1.46 hours. On the other hand, libvpx-vp9 does not use the CPU’s capacity efficiently, exploiting, on average, only 40% of the total computing capacity. Therefore, although libvpx-vp9 does not use CPU capacity well, it gives less encoding time than x265 and VVenC. Regarding memory usage, we found that libvpx-vp9 doesn’t need high memory to encode a video with a slower preset compared to other codecs since it only uses 0.96% of memory on average. We also observe that the most memory-hungry codec for a slower preset is SVT-AV1, which uses, on average, 7.05% of the total memory to encode a video. From Figure 9(b), we can observe that memory utilization differs between faster and slower presets. Compared to the slower preset, it is apparent that memory usage is significantly reduced for x264, x265, SVT-AV1, and VVenC. However, for libvpx-vp9 there is an increase in memory usage. Moreover, it is worth mentioning that for a faster preset, x265 is the most memory-efficient encoder.

## CONCLUSION AND DISCUSSIONS

In this work, we provided an extensive comparative study of energy consumption and CO2 emissions for leading video coding standards, aligning the results



**Figure 9.** Average memory, CPU usage, and the total encoding time (run time) of JVET-CTC dataset across five considered encoders.

with video quality and bitrate. The study encompassed five popular open-source and optimized video encoders, namely x264, x265, VVenC, libvpx9, and SVT-AV1, using four different encoder presets. The results on the JVET-CTC dataset showed that each newly born encoder outperforms other older encoders in terms of coding efficiency at the expense of an increase in complexity. The best coding efficiency is provided by VVenC based on wPSNR, SSIM, and VMAF quality metrics. Regarding energy consumption and CO<sub>2</sub> emissions, the obtained results illustrated that the encoders consume high energy when using the slower preset. For the four presets, VVenC exhibits the highest energy consumption and CO<sub>2</sub> emissions. In contrast, the x264 encoder provides the most negligible energy consumption and CO<sub>2</sub> emissions, making it the most energy-efficient encoder, while the SVT-AV1 encoder enables the best trade-off between coding efficiency and energy consumption. Notably, the encoder, preset, and resolution significantly impact energy consumption and CO<sub>2</sub> emissions. This study carries profound implications for both video streaming industry stakeholders and consumers, furnishing them with invaluable insights to craft directives and adopt optimal strategies for curtailing and mitigating their carbon impact. It underscores the significance of transparently disclosing carbon emissions to end-users and streamlining encoding and transcoding processes. This optimization should encompass factors like geographical location, codec selection, and encoding parameters

such as resolution, preset, and bitrate, while also accounting for the resultant quality and latency. Armed with the insights and recommendations from this study, video streaming entities have the potential to chart a course towards more sustainable practices within the industry, thus making a substantial contribution to the broader endeavor of environmental impact reduction.

In addition, it would be very interesting to study the coding efficiency and energy consumption of hardware encoders, which is one of the future works to be addressed.

## ACKNOWLEDGMENTS

This work was partially supported by the Hubert Curien Partnerships (PHC) Maghreb 2021, No 45988WG (Eco-VVC project) and the NESTED project, funded by the Brittany Region and Rennes Métropole.

## REFERENCES

1. E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Jul. 2019.
2. K. Lottick, S. Susai, S. Friedler, and J. Wilson, "Energy Usage Reports: Environmental awareness as part of algorithmic accountability," in *NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning*, 2019.
3. A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres,

- “Quantifying the carbon emissions of machine learning,” in *NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning*, 2019.
4. P. Li, J. Yang, M. A. Islam, and S. Ren, “Making AI Less “Thirsty”: Uncovering and Addressing the Secret Water Footprint of AI Models,” *arXiv preprint arXiv:2304.03271*, 2023.
  5. Greening of Streaming.  
<https://www.greeningofstreaming.org>. Accessed on 2-June-2023.
  6. Sandvine, “Global internet phenomena report 2022,” <https://www.sandvine.com/phenomena>, accessed on 30-January-2023.
  7. V.-V. Mattila, “Immersion, interactivity, cloudification, m2m and ai: 5 trends in future video usage,” in <https://www.bell-labs.com/institute/blog/immersion-interactivity-cloudification-m2m-and-ai-5-trends-future-video-usage/>, June 2020.
  8. M. Efoui-Hess, “Climate Crisis: The Unsustainable Use of Online Video, The practical case for digital sobriety,” in *The Shift Project*: <https://theshiftproject.org/wp-content/uploads/2019/07/2019-02.pdf>, July 2019.
  9. B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, “Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC),” *Proc. IEEE*, vol. 109, no. 9, pp. 1463–1493, 2021.
  10. T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.
  11. 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, 2012.
  12. W. Hamidouche *et al.*, “Versatile Video Coding Standard: A Review from Coding Tools to Consumers Deployment,” *IEEE Consum. Electron. Mag.*, 2022.
  13. B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, “Overview of the Versatile Video Coding (VVC) Standard and its Applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, 2021.
  14. C. Bonnineau *et al.*, “Perceptual Quality Assessment of HEVC and VVC Standards for 8K Video,” *IEEE Trans. Broadcast.*, vol. 68, no. 1, pp. 246–253, 2022.
  15. D. Grois *et al.*, “Performance Comparison of Emerging EVC and VVC Video Coding Standards with HEVC and AV1,” *SMPTE Motion Imaging Journal*, vol. 130, no. 4, pp. 1–12, 2021.
  16. J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, “Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, 2012.
  17. F. Bossen, K. Sühring, A. Wiecekowsky, and S. Liu, “VVC Complexity and Software Implementation Analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3765–3778, 2021.
  18. Z. Li, Z. Duanmu, W. Liu, and Z. Wang, “AVC, HEVC, VP9, AVS2 OR AV1?—A comparative study of state-of-the-art video encoders on 4K videos,” in *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019*, pp. 162–173.
  19. M. Uitto, “Energy consumption evaluation of H.264 and HEVC video encoders in high-resolution live streaming,” in *IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016, pp. 1–7.
  20. D. Silveira, M. Porto, and S. Bampi, “Performance and energy consumption analysis of the X265 video encoder,” in *25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1519–1523.
  21. R. Mathá, D. Kimovski, A. Zabrovskiy, C. Timmerer, and R. Prodan, “Where to Encode: A Performance Analysis of x86 and Arm-based Amazon EC2 Instances,” in *IEEE 17th International Conference on eScience (eScience)*, 2021, pp. 118–127.
  22. G. Ramasubbu, A. Kaup, and C. Herglotz, “Modeling the HEVC Encoding Energy Using the Encoder Processing Time,” in *IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 3241–3245.
  23. A. Katsenou, J. Mao, and I. Mavromatis, “Energy-Rate-Quality Tradeoffs of State-of-the-Art Video Codecs,” in *Picture Coding Symposium (PCS)*, 2022, pp. 265–269.
  24. H. Amirpour, V. V. Menon, S. Afzal, R. Prodan, and C. Timmerer, “Optimizing Video Streaming for Sustainability and Quality: The Role of Preset Selection in Per-Title Encoding,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2023.
  25. M. Uitto and M. Forsell, “Towards Energy-Efficient Adaptive Mpeg-Dash Streaming Using Hevc,” in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2018.
  26. A. Saha *et al.*, “Performance analysis of optimized versatile video coding software decoders on embedded platforms,” *Journal of Real-Time Image Processing*, vol. 20, no. 120, 2023.



27. M. Kränzler, C. Herglotz, and A. Kaup, "Decoding Energy Modeling For Versatile Video Coding," in *IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3144–3148.
28. M. Kränzler *et al.*, "Optimized Decoding-Energy-Aware Encoding In Practical VVC Implementations," in *IEEE International Conference on Image Processing (ICIP)*, 2022.
29. C. Herglotz, D. Springer, M. Reichenbach, B. Stabernack, and A. Kaup, "Modeling the Energy Consumption of the HEVC Decoding Process," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 217–229, 2018.
30. C. Herglotz, A. Heindel, and A. Kaup, "Decoding-Energy-Rate-Distortion Optimization for Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 171–182, 2019.
31. T. Chachou, W. Hamidouche, S. A. Fezza, and G. Belalem, "Energy consumption and carbon footprint of modern video decoding software," in *IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2023.
32. Z. Yan and C. W. Chen, "RnB: Rate and brightness adaptation for rate-distortion-energy tradeoff in HTTP adaptive streaming over mobile devices," in *Proc. of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 308–319.
33. "ISO/IEC 23001-11:2023 Information technology — MPEG systems technologies — Part 11: Energy-efficient media consumption (green metadata)," 2023, accessed on 29-March-2023. [Online]. Available: <https://www.iso.org/standard/83674.html>
34. A. Saha, W. Hamidouche, M. Chavarrías, F. Pescador, and I. Farhat, "Performance analysis of optimized versatile video coding software decoders on embedded platforms," *Journal of Real-Time Image Processing*, vol. 20, no. 6, p. 120, 2023.
35. M. Kränzler, C. Herglotz, and A. Kaup, "A Comparative Analysis of the Time and Energy Demand of Versatile Video Coding and High Efficiency Video Coding Reference Decoders," in *IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020.
36. M. Kränzler, C. Herglotz, and A. Kaup, "Energy Efficient Video Decoding for VVC Using a Greedy Strategy-Based Design Space Exploration," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4696–4709, 2022.
37. F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "JVET common test conditions and software reference configurations for SDR video," *Joint Video Experts Team (JVET) of ITU-T SG*, vol. 16, pp. 19–27, 2019.
38. P.-H. Wu, V. Kondratenko, and I. Katsavounidis, "Fast encoding parameter selection for convex hull video encoding," in *Applications of Digital Image Processing XLIII*, vol. 11510. SPIE, 2020, pp. 181–194.
39. Electricity Maps. <https://app.electricitymaps.com/map>. Accessed on 4-March-2022.
40. Reading RAPL energy measurements from Linux. <https://web.eece.maine.edu/~vweaver/projects/rapl/index.html>. Accessed on 19-September-2022.
41. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
42. Z. Li *et al.*, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, vol. 6, no. 2, p. 2, 2016.
43. G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, 2001.
44. European Environment Agency. <https://www.eea.europa.eu/data-and-maps/indicators/average-co2-emissions-from-motor-vehicles/assessment-1>. Accessed on 5-October-2022.
45. P. ITU-T RECOMMENDATION, "Subjective video quality assessment methods for multimedia applications," 1999.
46. V. V. Menon, C. Feldmann, H. Amirpour, M. Ghanbari, and C. Timmerer, "VCA: Video complexity analyzer," in *Proc. of the 13th ACM multimedia systems conference*, 2022, pp. 259–264.

**Taieb Chachou** is finishing his PhD at the Université Oran1, Algeria. Contact him at [taieb.chachou@gmail.com](mailto:taieb.chachou@gmail.com).

**Wassim Hamidouche** is currently a principal researcher at the Technology Innovation Institute (TII) in Abu Dhabi, UAE. He also holds the position of associate professor at INSA Rennes, France. Contact him at [wassim.hamidouche@insa-rennes.fr](mailto:wassim.hamidouche@insa-rennes.fr).

**Sid Ahmed Fezza** is a full professor at the National Higher School of Telecommunications and ICT, Algeria. Contact him at [sfezza@ensttic.dz](mailto:sfezza@ensttic.dz).

**Ghalem Belalem** is a professor with the Department of Computer Science, Université Oran1, Algeria. Contact him at [ghalem1dz@gmail.com](mailto:ghalem1dz@gmail.com).