Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

Performance Evaluation of AVC and HEVC for E-Learning: Optimizing Quality and Reducing Bandwidth Usage

Oğuz KIRAT[1*], Tarık YERLİKAYA[2], Emir ÖZTÜRK[3]

[1] Trakya University, Computer Engineering Department, oguzkirat@trakya.edu.tr, Orcid No: 0000-0003-2687-9351
[2] Trakya University, Computer Engineering Department, tarikyer@trakya.edu.tr, Orcid No: 0000-0002-9888-0151
[3] Trakya University, Computer Engineering Department, emirozturk@trakya.edu.tr, Orcid No: 0000-0002-3734-5171

ARTICLE INFO

Article history:

Received 31 July 2023
Received in revised form
28 November 2023
Accepted 5 December 2023
Available online 31 December 2023

Keywords:

AVC, Compression, E-Learning, HEVC, Video Encoding, Video Quality Assessment

Doi: 10.24012/dumf.1335369

* Corresponding author

ABSTRACT

E-learning has experienced a surge in popularity, particularly during and after the COVID-19 pandemic. Online learning has proven to be a vital tool for students and educators to continue academic activities while adhering to social distancing guidelines and during the times of natural disasters that disrupt the conventional learning environments. It also offers accessibility to disabled students and those facing challenges to reach to the traditional learning. But due to increased demand, it is crucial to optimize cost of transmission while minimizing bandwidth usage while maintaining high-quality video transmission. To optimize cost and reduce network load, it is essential to minimize bandwidth usage while maintaining high-quality video. In response to this need, we present a novel dataset consisting of four e-learning scenarios. We encoded this dataset using various resolutions, bit rates, and encoder presets, and evaluated it in terms of encoding time, and quality using full-reference objective quality metrics such as MSE, PSNR, and SSIM. After experimenting with more than 1400 videos and configurations of encoders and codecs, we found out that it is possible to transmit videos in exceptional quality at bit rates as low as 5 Mbps for e-learning scenarios. We also present detailed results about correlation between file size, quality and encoding time to make optimizations for specific bandwidth, target quality or encoding speed.

Introduction

According to CISCO Visual Networking Index, video streaming accounts for 82% of the global internet traffic. For 2022, it is predicted to be between 80% and 90%, surpassing the expectations of previous years [1]. Live streaming, on the other hand, accounts for 17% of internet traffic. During the COVID-19 pandemic, there was an unprecedented surge in the popularity of online lessons, virtual meetings, and e-learning platforms. As a result, an ever-increasing abundance of course materials are now accessible on the internet, supplementing the pre-existing digital resources.

The efficient transmission of multiple videos over the same network connection and cost of server and datacenter equipment necessitates a proficient utilization of network bandwidth at the source. Meanwhile, users on the receiving end may contend with restricted network connections, characterized by both limited latency and bandwidth, making effective video compression paramount. To cater to these requirements, various video encoding and compression standards have been developed, accommodating both lossy and lossless compression options. However, for end users, lossy compression is typically employed, as video files compressed with lossless methods often entail massive file sizes.

One consequence of employing lossy compression is the potential presence of distortions in the video output. In the context of e-learning videos, it becomes imperative to strike a careful balance in the choice of the compression rate. This is crucial because certain minute details, such as individual letters and intricate figures, might hold vital importance. Consider, for instance, the readability of programming code in a tutorial – a facet that must not be compromised. Unlike typical movies, our dataset primarily comprises relatively static subjects, minimizing complex scenarios that might pose challenges for video encoders. For instance, we encounter few instances of rapidly changing scenes, camera shake effects, explosive particles, or film grain effects, which are often difficult to compress optimally.

In light of these considerations, it is evident that the optimization of video compression techniques plays a pivotal role in ensuring the seamless delivery of educational content in the realm of e-learning. By fine-tuning the compression parameters to strike an ideal balance between file size and visual fidelity, educators, content creators, e-learning platforms and network operators can guarantee an enhanced learning experience for their audiences.

In recent years, various codecs have been developed, such as VP9 [2][3], AV1, VVC [4], ELF-VC [5], and LC-EVC [6]. Though some of them provide more compression in exchange for processing power, hardware encoder/decoder support needs to be considered when the aim is to reach as many users as possible. Therefore, we used H.264 [7] and H.265 [8] codecs in this paper.

H.264 was presented by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC JTC1 Moving Picture Experts Group (MPEG) in 2003 and is widely used as the default video encoding standard on many platforms. Although there has been a shift towards alternative applications in recent years due to the increase in video resolutions and bit rates with bandwidth, it has not yet lost its popularity. However, some limitations imposed at the time of the standard's release prevent certain features from being added, hence the need for new methods.

H.265, similarly introduced by VCEG and MPEG in 2013, offers significant improvements over H.264 and presents various mechanisms and improvements to achieve the same quality at higher resolutions with smaller file sizes. Ohm and colleagues, in their study, found that H.265 can reduce the bit rate by 50% compared to H.264. Recently, many hardware manufacturers have also provided hardware-level encoding support for H.265 [9].

Previous studies have compared and measured the effectiveness of lossy coding methods from different perspectives.

Guo et al. compared movies and series on streaming service platforms using the H.264, H.265, and VP9 encoding methods and found that H.265 yielded better results than VP9[10]. They also observed that the difference between encoding methods decreased with an increasing resolution.

Barman and Martini [11] compared H.264, H.265, and VP9 encoding methods on selected gaming videos on the Twitch platform and argued that H.264 was much faster and more efficient in terms of bit rate than VP9. They also stated that the VP9 encoding method gave different results depending on the content.

Calı and Özbek encoded specific images at fixed and variable bit rates and evaluated performance during transmission using Dynamic Adaptive Streaming over HTTP (DASH) [12].

Kılıç and colleagues compared the compression rate, quality, and encoding time of new-generation video encoding methods for sports, news, and animation videos and obtained quality loss using VMAF, PSNR, and SSIM [13]. The study found that the AV1 and VP9 approaches produced much better results than H.264 but particularly highlighted that the AV1 encoding method had a very high hardware requirement.

Nguyen and his colleagues compared the VVenC, x265, and AV1 encoding methods on videos containing street and road images and analyzed the results regarding PSNR, XPSNR, MS-SSIM, and encoding times [14].

In prior research, studies predominantly focused on encoding scenarios involving movies, series, and game streams, where foreground objects were the primary subject of interest. There are some metrics specific to the type of content [15][16]. However, a significant gap exists in the literature concerning the comparison of results for images containing text-weighted windows. To address this limitation, our study sought to experiment on a specific dataset that emphasizes different aspects compared to previous investigations.

In contrast to other studies, we specifically examined the impact of video compression on text elements as distinct from its effects on objects and scenes. Interestingly, blurring certain portions within objects did not significantly hinder the discernibility of those objects. However, when it came to text, any form of blurring noticeably affected the readability and legibility of the textual content.

In this study, we created a set of four e-learning scenarios that are exactly 30 seconds long. We subsequently encoded them using both the H.264 and H.265 codecs with various configurations. Through these encoding tests, we measured the resulting file sizes, quantified the quality ratios in terms of Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) [17], and calculated the encoding times. These essential data points enabled us to identify the most suitable encoding method to employ under specific bandwidth constraints.

By conducting this comprehensive investigation, we aim to fill the gap in knowledge regarding the optimal video compression techniques for e-learning scenarios, particularly when text plays a crucial role in the conveying information.

Material and Method

For the study, we created a novel dataset consisting of four videos each representing a different scenario. For all videos, we captured lossless screen capture and webcam input. Every video is precisely 30 seconds long with a resolution of 1920x1080 pixels at 30 frames per second. The dataset is available on request.

For every scenario, webcam input was positioned at the bottom right corner of the screen capture for the inclusion of a speaking presenter. This configuration is set up to simulate a real-time or recorded online class experience, enhancing the relevance of the dataset.

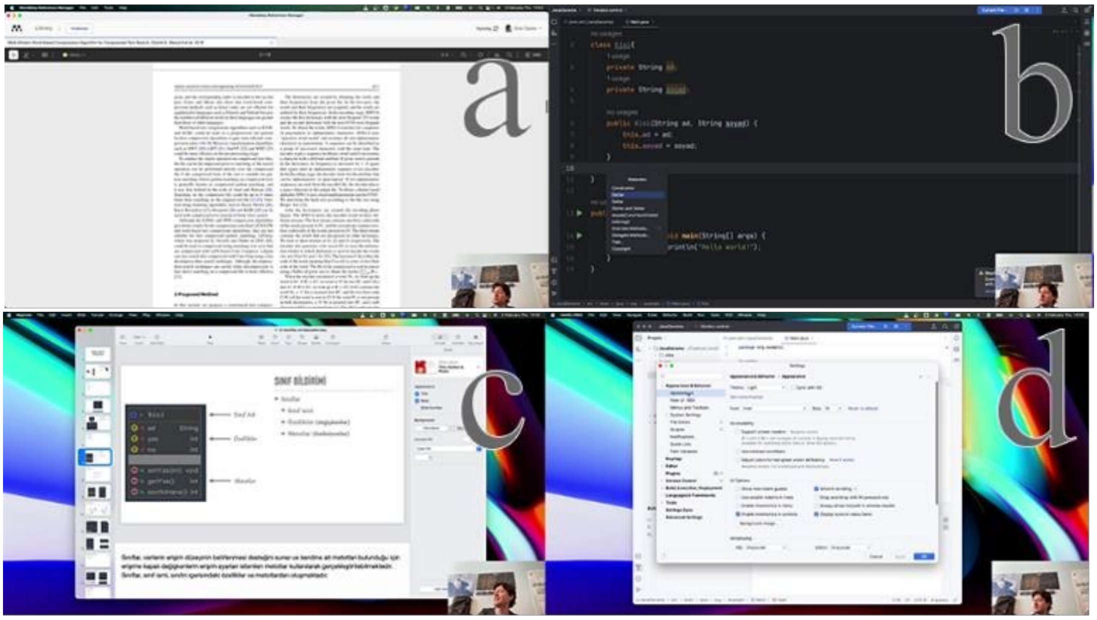

Figure 1. Frame samples: Article (a) Coding (b) UI (c) and Presentation (d) scenario

Four scenarios that are inspected in this study are explained in Table 1.

Table 1. Explanation of the videos in the dataset.

Scenario	Description
Presentation	The lecturer presents a course presentation. Content is generally static.
User Interface	The lecturer talks about configuring a program on its user interface.
Article Review	The lecturer presents a PDF article. The article was scrolled and zoomed in during the video. Therefore, this is the most complex scenario.
Coding	The lecturer writes a Java class on an IDE.

Sample frames from each video scenario in Table 1 are given in Figure 1. As seen in Figure 1, each scenario contains different types and amounts of text. We also downsampled the lossless 1920x1080 pixel resolution videos from each scenario to 1280x720 pixels, using bicubic interpolation to see the effects of reduction in resolution.

While some video streaming services may offer 4K resolution option, it is pertinent to note that 4K streaming is limited to specific devices and demands high bandwidth requirements. As a result, for the purpose of this study, we have deliberately excluded resolutions exceeding 1920x1080 pixels. H.264 and H.265 are selected due to their high compatibility and large number of supported hardware decoders.

In order to account for diverse network technologies and their associated bandwidth limitations, we considered a range of bit rates for our experiments. These bit rates were specifically chosen to span from UMTS/R99, which provides a maximum bandwidth of 384 kbps, to broadband connections offering up to 20 Mbps. This approach enables us to assess the performance of the chosen video codecs across various network conditions, ensuring that our conclusions remain applicable to real-world scenarios with varying network capacities.

Table 2 highlights the various configuration options explored in our study, encompassing all the cases used in our experiments.

Table 2. All configuration options in the study.

Configuration Type	Possibilities Tested
Scenarios	Article Review, Coding, Presentation, User Interface
Codecs	H.264 (AVC), H.265 (HEVC)
Resolutions	1280x720, 1920x1080 pixels
Presets	veryslow, slower, slow, medium, fast, faster, veryfast, superfast, ultrafast
Bit rates	300 kbps, 500 kbps, 700 kbps, 1 Mbps, 2 Mbps, 5Mbps, 7 Mbps, 10 Mbps, 15 Mbps, 20 Mbps

We encoded all videos with different scenarios for each codec and resolution using nine different presets and ten different bit rates. As a result, we created 1440 different output video samples and calculated objective metrics for each video output.

The presets for x264 and x265 are named *as ultrafast, superfast, veryfast, faster, fast, medium, slow*, *slower* and *veryslow*. In video encoding, the encoder takes numerous different parameters. Modifying these parameters can affect the encoding and decoding speed, the quality of the encoded video, and the output video size. Each given preset modifies these specific parameters for the encoding method. These parameters mostly involve additional operations during the encoding process to enhance the quality at similar file size. However, in return, they increase the processing complexity, leading to lower encoding and decoding speeds.

For example, *bframes* flag in encoding method determines the number of B frames in encoding process. In video encoding process, there are three types of frames named I, P and B frames and B frames are encoded bidirectionally using frames in both directions. These are the frames with the highest processing complexity to decode. Increasing the number of B frames will increase the compression ratio as well as the processing complexity.

As moving from the *medium* preset to the *veryslow* preset, quality is increased with the increasing number of operations on frames, using estimation algorithms etc., but each additional operation has a negative effect on the encoding time.

All conversions are made using x264 AVC encoder version 0.164.3095baee400 and x265 HEVC encoder version 3.4+31-6722fce1f

MSE, PSNR, and SSIM are used for quality measurement. MSE is a metric for calculating error (noise/degradation) by comparing images pixel by pixel. Formula for MSE is given in equation 1.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m}\sum_{j=0}^{n}||f(i,j) - g(i,j)||^2 \qquad (1)$$

where f is the original image matrix, g is matrix of the image to be compared, m is the number of rows of pixels, and i is the row index. Similarly, n is the number of columns of pixels and j is the index of that column. MSE values close to 0 is better.

PSNR (Peak Signal-to-Noise Ratio) is logarithmic scale of noise in a signal, and widely used in images. If there is no signal, thus two images identical PSNR value is infinite. When the noise increases, PSNR decreases.

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right) \qquad (2)$$

R is the maximum fluctuation in the input image data type. For example, if an image has an 8-bit unsigned integer data type, R is 255.

As stated by Kufa and Kratochvil, the PSNR value is expected to increase in parallel to quality [18].

SSIM is a metric to compare similarity between two images where μ_x, μ_y, σ_x,σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images x, y.

$$SSIM(i,j) = \frac{(2\mu_i\mu_j + c_1)(2\sigma_{ij} + c_2)}{(\mu_i^2 + \mu_j^2 + c_1)(\sigma_i^2 + \sigma_j^2 + c_2)} \qquad (3)$$

If SSIM value is 1, images perfectly match each other. For quality assessment, we calculated average MSE, PSNR and SSIM values of frames using a function given below:

$$Q_f(o,d) = \frac{\sum_{k=1}^{n}\frac{f_y(o_k,d_k) + f_u(o_k,d_k) + f_v(o_k,d_k)}{3}}{n} \qquad (4)$$

The parameters and the descriptions of the equation are given in Table 3.

Table 3. Parameters of metric calculation.

Parameter	Explanation
Q	Average representation of the metric for encoded video
o	Original video
d	Compressed/degraded video
f_y	Metric function for Y channel (MSE, PSNR or SSIM)
f_u	Metric function for U channel (MSE, PSNR or SSIM)
f_v	Metric function for V channel (MSE, PSNR or SSIM)
n	Number of frames in a video (900 for our dataset)
k	k^{th} frame of the video

Experimental Results and Discussion

The video encoding process, especially when performed on software-based solutions, is known to be highly CPU intensive. Consequently, employing unnecessarily slow encoder configurations can lead to a significant waste of time and energy, especially if the resulting video quality does not show substantial improvement. To determine the optimal encoding preset for e-learning purposes, we embarked on an extensive encoding process on our dataset, generating a total of 1440 output videos.

Given that the internet might be shared among multiple users in the e-learning environment, the receivers' (students') internet connection speed could be limited, or lecturers' upload speed might be even slower. Therefore, we conducted encoding tests with a range of bit rates. These bit rates started from 300 kbps and reached up to 20 Mbps. This approach allows us to explore the performance of the codecs under various network conditions and bandwidth limitations.

To execute these encoding tests, we utilized a test machine featuring a water-cooled AMD Ryzen 5 5600 processor, which has six cores and twelve threads. This CPU has base clock speed of 3.5 GHz and boost clock speed up to 4.4 GHz. The processor was installed on a B450 chipset motherboard, and the test system was equipped with 32 GBs of 3600 MHz DDR4 RAM. To ensure the accuracy and reliability of the results, we ensured the system operated over the base clock speeds and was not thermally throttled during the tests.

Detailed experimental results are given in the subsections below.

Encoding Speed vs. Presets

In our test process we utilized the x264 and x265 encoders. Both encoders provide flexibility of selecting different presets, giving us the opportunity to tweak the balance between encoding speed and compression efficiency. The default preset for both encoders is *medium*. However, when faster presets are chosen, certain parameters are adjusted to sacrifice video quality in favor of increased encoding performance. Conversely, selecting slower presets prompts the codec to employ additional operations, thus using more processing power and time, but in exchange enhancing the video quality.

For AVC (H.264) encoding using x264, we observed that the *medium*, *fast*, and *faster* presets demonstrated similar encoding times. However, the *slow* preset had a minimum encoding time that is nearly identical to the average time of the *medium* preset. But the encoding time gap widened when comparing the *slow*, *slower* and *veryslow* presets. Additionally, we noticed considerable variation in encoding times within the same preset, especially while using the *slow*, *slower* and *veryslow* presets, depending on the specific scenario and bit rate.

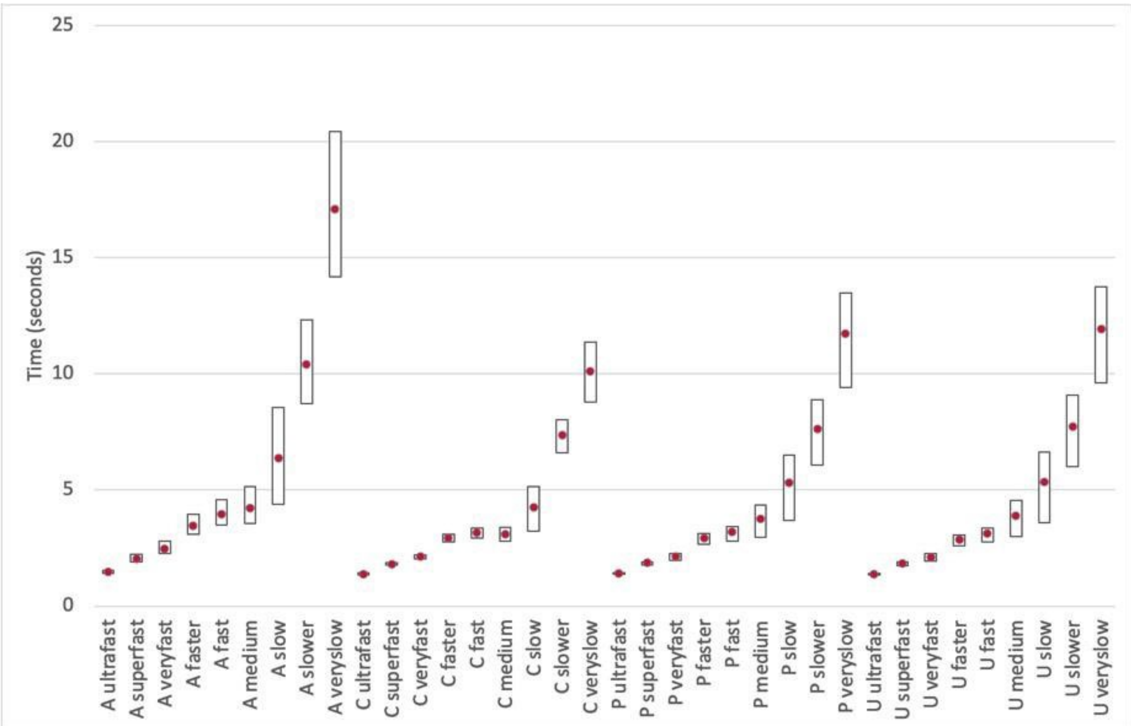

Figure 2. H.264 (AVC) encoding using x264, average encoding time range for each scenario (A=Article, C=Coding, P=Presentation, U= User Interface) and preset, all bit rates combined. The red dots represent average points.

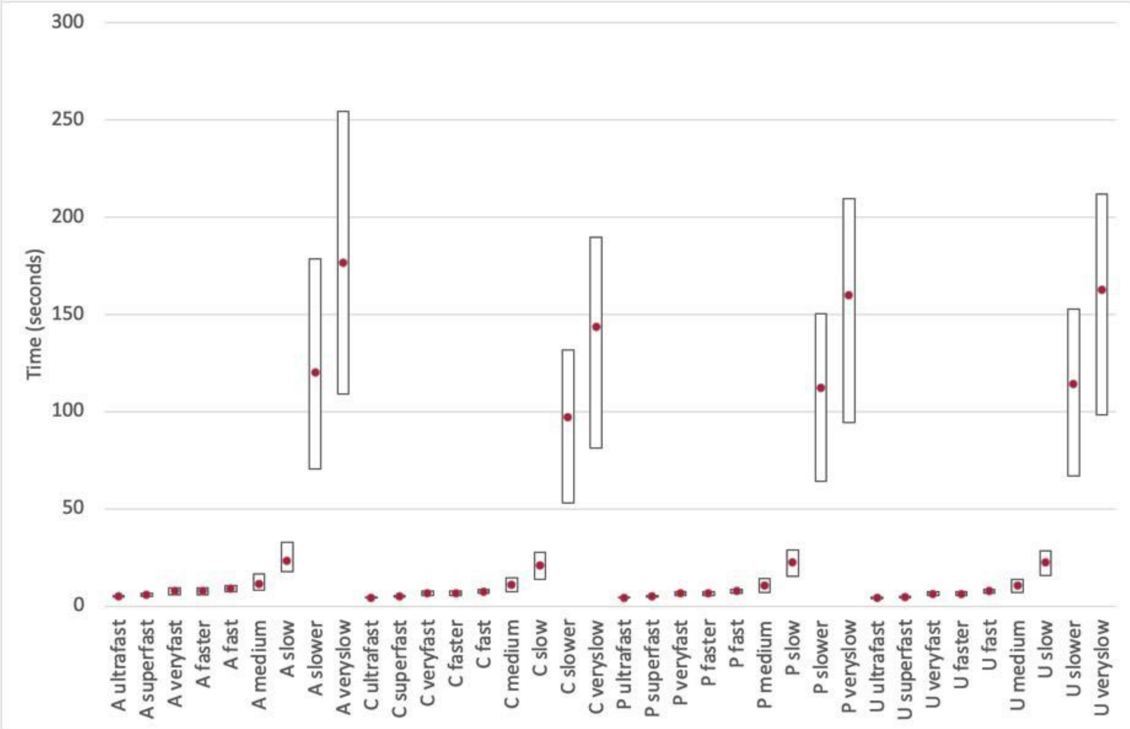

Figure.3. H.265 (HEVC) encoding using x265, average encoding time range for each scenario (A=Article, C=Coding, P=Presentation, U= User Interface) and preset, all bit rates combined. Red dots represent the average point.

In contrast to H.264 encoding using x264, when we applied H.265 encoding using x265, we observed a more pronounced difference in encoding times between the *slower* and *slow* presets. The *slower* preset exhibited encoding times that are between 4-5 times slower than the *slow* preset in most encoding scenarios, and the *slow* preset generally doubled the encoding time required by the *medium* preset. These substantial differences indicate that selecting presets slower than the *medium* preset might not be the most optimal choice considering encoding time, especially if the *medium* preset can deliver the expected video quality for the given scenario. Additionally, we found that encoding videos with a resolution of 1280x720 pixels took less time than encoding videos at 1920x1080 pixel resolution.

Table 4 presents the correlation between preset and encoding time, with the time values representing the averages across all scenarios and bit rates.

From the results, it is evident that the article scenario took the longest time to encode, which was expected due to the motion caused by scrolling and zooming in the video. Conversely, the coding scenario took the shortest time to encode. Figure 2 (for H.264) and Figure 3 (for H.265) further illustrate the encoding time ranges for each scenario and preset.

Table 4. Correlation between preset and encoding time.

Codec / Res.	Preset	Avg. Time (s)	Codec / Res.	Preset	Avg. Time (s)
H.264 1080p	ultrafast	1.41	H.265 1080p	ultrafast	4.40
	superfast	1.89		superfast	5.03
	veryfast	2.22		veryfast	6.76
	faster	3.05		faster	6.77
	fast	3.36		fast	7.97
	medium	3.74		medium	10.78
	slow	5.33		slow	22.25
	slower	8.29		slower	110.90
	veryslow	12.72		veryslow	160.73
H.264 720p	ultrafast	0.78	H.265 720p	ultrafast	2.17
	superfast	0.99		superfast	2.74
	veryfast	1.12		veryfast	4.37
	faster	1.49		faster	4.37
	fast	1.65		fast	5.18
	medium	1.91		medium	6.45
	slow	2.86		slow	13.45
	slower	4.13		slower	72.77
	veryslow	6.54		veryslow	106.84

Regarding H.264 encoding, we observed that e-learning videos can be encoded in real-time on every preset using a modern CPU. For example, in the default *medium* preset, with the exception of two cases, every video could be encoded six times faster than its original length for every bit rate. However, for our test configuration with H.265, reaching real-time encoding on *slower* and *veryslow* presets proved to be unattainable.

Due to the low complexity of videos in our dataset, the *slow* preset for H.265 encoding could encode more than 30 frames per second (fps) for most configurations, especially at low bit rates. If real-time encoding is a requirement, the *medium* preset showed a minimum of 56.25 fps in the worst-case scenario and provided an average SSIM score of 1 for this specific case, which can be considered nearly lossless, thus might be recommended.

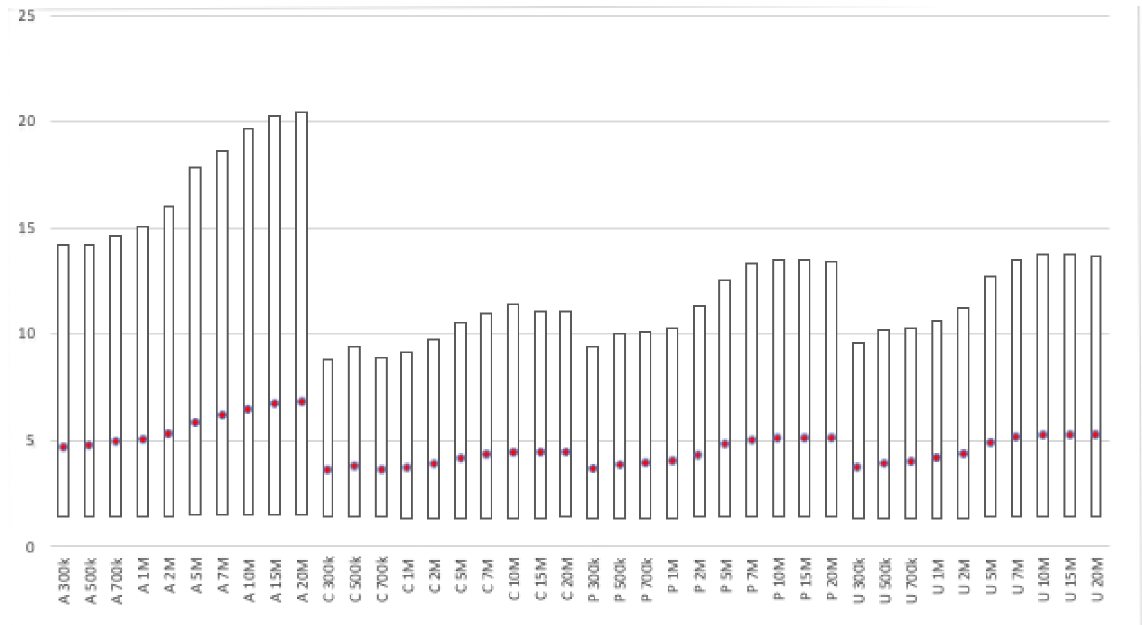

Figure 4. H.264 (AVC) encoding using x264, average encoding time range (s) for each scenario (A=Article, C=Coding, P=Presentation, U= User Interface) and bit rate, all presets combined. Red dots represent the average point.

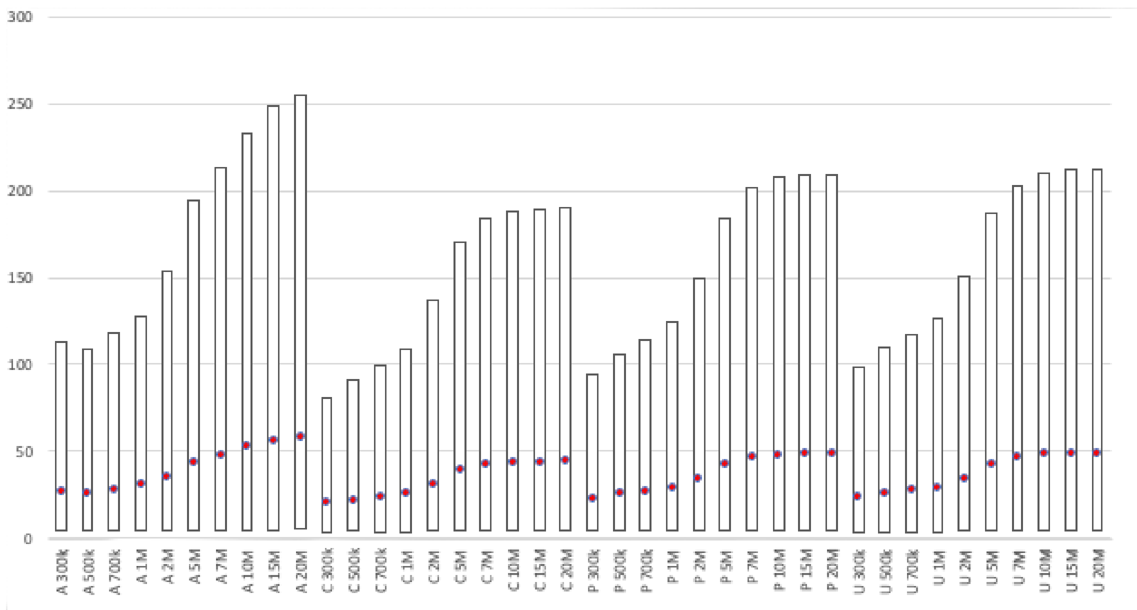

Figure 5. H.265 (HEVC) encoding using x265, average encoding time range (in seconds) for each scenario (A=Article, C=Coding, P=Presentation, U= User Interface) and bit rate, all presets combined. Red dots represent the average point.

Encoding Speed vs. Bit rate

When analyzing each scenario individually, the experimental results indicated that increasing the bit rate generally led to an increase in encoding time. However, it is important to note that the complexity of the video content directly influenced the encoding time.

Figure 4 and Figure 5 provide a detailed representation of the encoding time results for different bit rates across various scenarios.

As observed in the figures, the H.264 codec consistently achieved real-time encoding results for all bit rate configurations on a modern computer and using a modern encoder. H.265 codec, while performing efficiently at lower bit rates, started to experience a slowdown in encoding time after reaching the 5 Mbps mark. This suggests that beyond this threshold, the H.265 codec struggled to maintain real-time encoding capabilities.

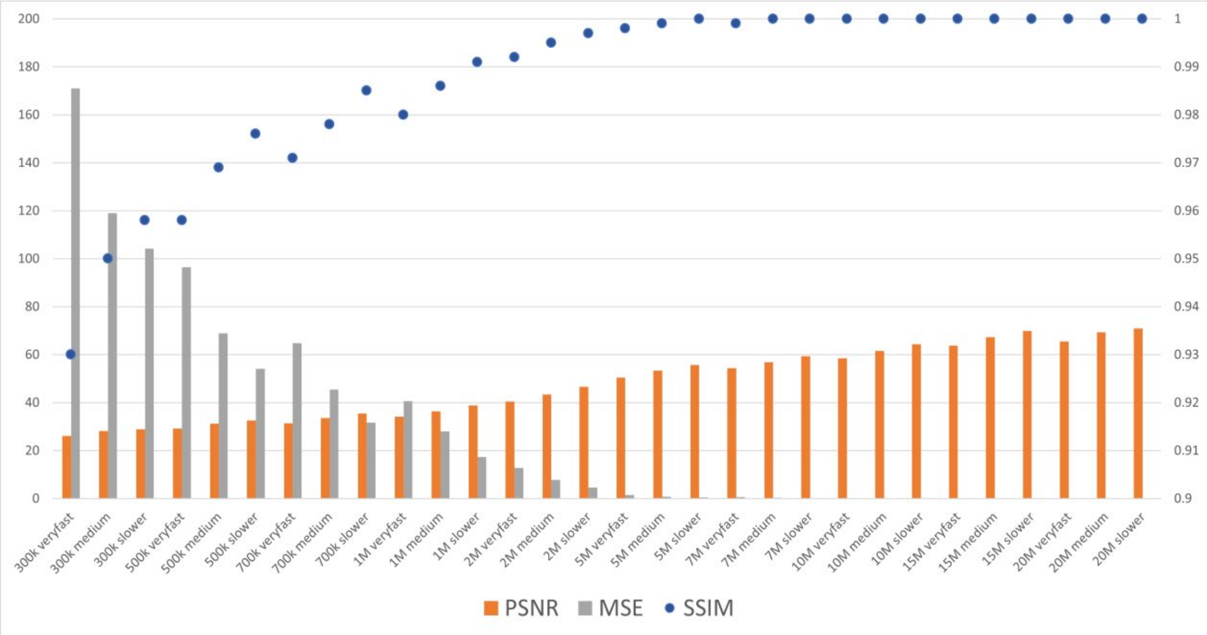

Figure 6. H.264 (AVC) encoding using x264 in 1920x1080 pixels, quality of article scenario.

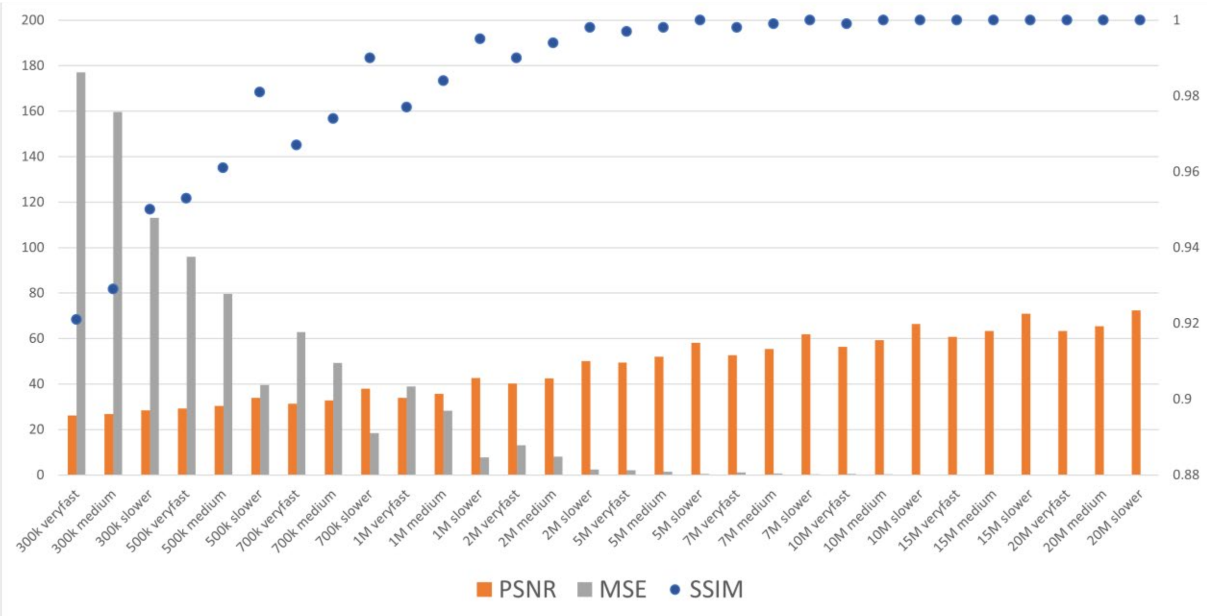

Figure 7. H.265 (HEVC) encoding using x265 in 1920x1080 pixels, quality of article scenario.

Encoding Time of AVC vs. HEVC

The experimental results unequivocally demonstrate that in nearly all cases, the encoding time required for HEVC (H.265) is significantly longer when compared to AVC (H.264). Based on our empirical data, it can be deducted that encoding videos using HEVC is at least 1.89 times slower than AVC, on average showing a difference of 5.78 times.

Particularly, in high bit rate scenarios and when utilizing slower encoding presets, the encoding time disparity between HEVC and AVC becomes more pronounced, up until HEVC being 21.82 times slower than AVC.

These findings demonstrate the importance of encoding times while selecting a codec, particularly in cases where hardware-based encoders are not usable.

Quality Comparison

In our effort to assess the quality of the encoded video outputs, we conducted a frame-by-frame comparison between each output and its corresponding original lossless source video. To ensure consistency and to evaluate Full HD display viewing, the 1280x720 resolution output videos were upscaled back to the original 1920x1080 pixel resolution using bicubic interpolation. It is important to note that due to the initial resolution decrease in the 1280x720 pixel resolution outputs, a maximum SSIM score of 0.983 is achieved within all outputs in that resolution. This outcome is anticipated, as some data loss occurs during the downscaling process.

For both H.264 and H.265 codecs, all configurations with bit rates greater than 2 Mbps achieved an SSIM greater than 0.99, indicating high visual fidelity and quality. While certain configurations of slower presets in H.264 and H.265 attained an SSIM value above 0.99, it is essential to acknowledge that this level of quality is not consistently achievable across all configurations.

In Figure 8, we present the calculated average SSIM values for each bit rate on the *veryfast* preset. Notably, among the HEVC (H.265) 1920x1080 pixel resolution video outputs, we obtained an average SSIM score of 1, which can be considered nearly lossless, for bit rates as low as 700 kbps for the *coding* scenario and 5 Mbps for the *article* scenario. These results demonstrate the impressive compression capabilities of the HEVC codec at relatively lower bit rates.

In Figures 6 and 7, we depict the calculated average quality metrics (PSNR and MSE) for the *article* scenario across different bit rates and presets (*veryfast*, *medium*, and *slower*). The left axis in the graphs represents PSNR and MSE values, while the right axis corresponds to the SSIM metric.

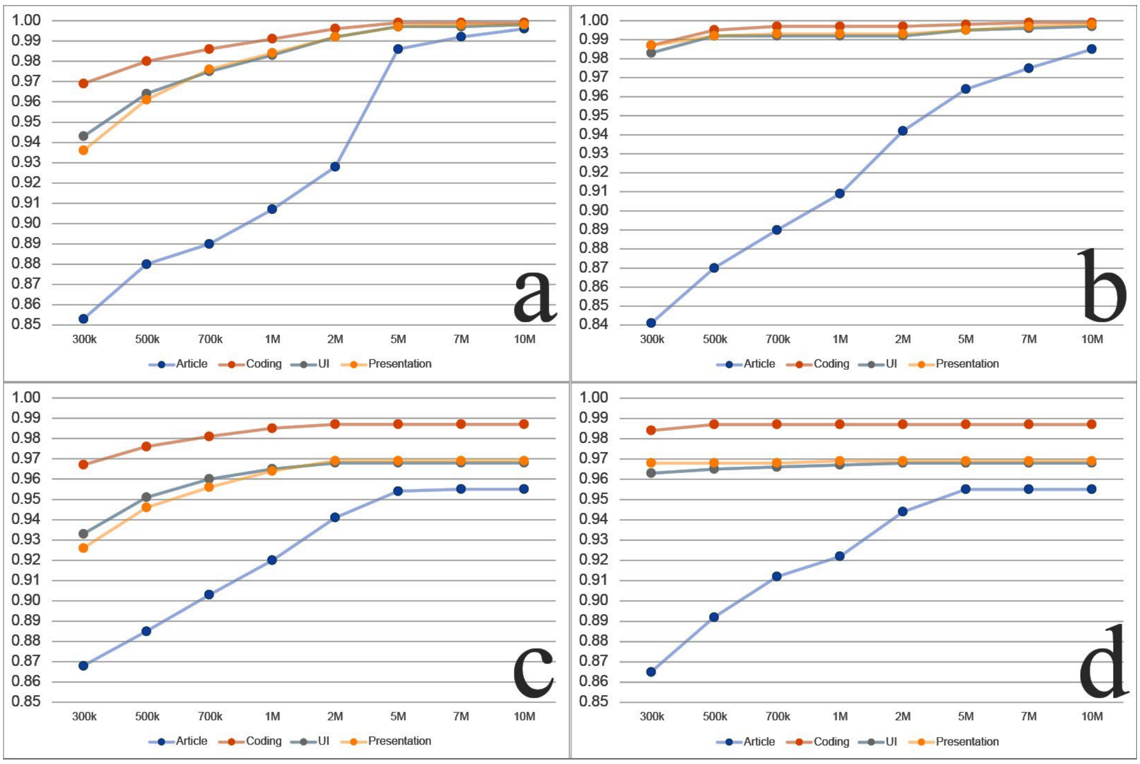

Figure 8. Average SSIM values for each bit rate on veryfast preset.
(a- AVC 1920x1080, b- HEVC 1920x1080, c- AVC 1280x720 d- HEVC 1280x720)

Conclusion

It is evident that in e-learning, videos primarily consist of text-based content, such as presentations and PDFs. Therefore, it is crucial to ensure that the encoding used during transmission over the network maintains a certain level of quality and readability for the text. Existing encoding methods are predominantly evaluated on datasets containing videos with mainly moving objects, and the evaluation of quality results is based on these datasets. However, in text-heavy videos, the blurring effect becomes more critical as it can significantly impact text legibility.

Based on the results obtained in the study, except for *slower* and *veryslow* preset configurations, both H.264 and H.265 codecs are capable of real-time encoding. H.264 shows real-time results across all the configurations, while H.265 starts to lose its real-time capability after reaching a 5 Mbps bit rate.

For both codecs, the study found that video quality after 2 Mbps is well-maintained without compromising text readability. Although the *slower* or *veryslow* codec configurations might offer better quality results than other presets, in some cases, it could be more practical to increase the bit rate instead of selecting these slower presets, as doing so would preserve the chance of real-time encoding.

The results of the study demonstrate that HEVC (H.265) generally outperforms AVC (H.264) in terms of average SSIM score. However, it is important to consider decoder availability and hardware decoding support. While HEVC offers higher compression efficiency and better quality, it comes with the trade-off of considerably slower encoding times, which might be a concern for real-time applications.

Even on AVC, it is noteworthy that selecting presets faster than *medium* with a bit rate between 2 Mbps and 5 Mbps can effectively transfer videos with sufficient quality for e-learning purposes. Moreover, an average bandwidth of 5 Mbps is adequate to transmit all videos at 1920x1080 pixel resolution, achieving SSIM scores near 1 and enabling near-real-time encoding on modern CPUs using x264.

The experiments conducted in this study have provided valuable insights into the impact of bit rate and preset selection on the output quality and processing time of video encoding. The findings hold significant implications for researchers and media producers, offering valuable guidance to improve their video encoding workflows since there is limited research on video quality assessment for e-learning.

In future studies, there are plans to develop a method for selecting the appropriate bit rate and encoding method dynamically based on the available bandwidth in real-time. This would involve examining the data obtained during transmission to ensure efficient utilization of available network resources while maintaining optimal video quality and readability for e-learning videos.

Ethics Committee Approval

There is no need to obtain permission from the ethics committee for the article prepared.

Conflict of Interest Statement

There is no conflict of interest with any person / institution in the article prepared.

Authors' Contributions

Each author contributed equally to the article prepared.

References

[1] T. Barnett, S. Jain, U. Andra, "Cisco visual networking index (vni) complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018, Accessed: Feb. 04, 2023. [Online]. Available: https://get.drivenets.com/hubfs/1211_BUSINESS_SERVICES_CKN_P DF.pdf

[2] "The WebM Project | Welcome to the WebM Project." https://www.webmproject.org/ (accessed Feb. 04, 2023).

[3] D. Mukherjee, J. Han, J. Bankoski, R. Bultje, A. Grange,J Koleszar, P. Wilkins, Y. Xu, "A technical overview of VP9 - The latest open-source video codec," *SMPTE Annual Technical Conference and Exhibition 2013, SMPTE 2013*, pp. 376–392, 2013, doi: 10.5594/M001518.

[4] Standard ISO/IEC 23090-3, *Versatile Video Coding*. 2020.

[5] O. Rippel, A. G. Anderson, K. Tatwawadi, S. Nair, C. Lytle, L. Bourdev, "ELF-VC: Efficient Learned Flexible-Rate Video Coding," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 14459–14468, Apr. 2021, doi: 10.48550/arxiv.2104.14335.

[6] Information Technology, "General Video Coding—Part 2: Low Complexity Enhancement Video Coding," *Standard ISO/IEC 230942:2021*. Nov. 2021.

[7] Recommendation ITU-T H.264 and ISO/IEC 14496-10 (AVC), "Advanced Video Coding for Generic Audio-Visual Services," *ITU-T and ISO/IEC JTC 1*. May 2003.

[8] Recommendation ITU-T H.265 and ISO/IEC 23008-2 (HEVC), "High Efficiency Video Coding," *ITU-T and ISO/IEC JTC 1*. Apr. 2013.

[9] G. J. Sullivan, J. R. Ohm, W. J. Han, T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649– 1668, 2012, doi: 10.1109/TCSVT.2012.2221191.

[10] L. Guo, J. de Cock, A. Aaron, "Compression Performance Comparison of x264, x265, libvpx and

aomenc for On-Demand Adaptive Streaming Applications,” *2018 Picture Coding Symposium, PCS 2018 - Proceedings*, pp. 26–30, Sep. 2018, doi: 10.1109/PCS.2018.8456302.

[11] N. Barman, M. G. Martini, “H.264/MPEG-AVC, H.265/MPEG-HEVC and VP9 codec comparison for live gaming video streaming,” *2017 9th International Conference on Quality of Multimedia Experience, QoMEX 2017*, Jun. 2017, doi: 10.1109/QOMEX.2017.7965686.

[12] M. Cali, N. Ozbek, “Comparative analysis of adaptive streaming with CBR and VBR encoded scalable videos,” *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications, Proceedings*, Jun. 2021, doi: 10.1109/SIU53274.2021.9477909.

[13] T. Kilic, A. M. Gulbahce, M. Yildiz, “Performance Evaluation of Next
Generation Video Compression Approaches Using Objective Quality Measurement Methods,” *2020 28th Signal Processing and Communications Applications Conference, SIU 2020 - Proceedings*, Oct. 2020, doi: 10.1109/SIU49456.2020.9302130.

[14] T. Nguyen, A. Wieckowski, B. Bross, D. Marpe, “Objective Evaluation of the Practical Video Encoders VVenC, x265, and aomenc AV1,” *2021 Picture Coding Symposium, PCS 2021 - Proceedings*, Jun. 2021, doi: 10.1109/PCS50896.2021.9477482.

[15] Z. Shang, Y. Chen, Y.Wu, H. Wei, S. Sethuraman “Subjective and Objective Video Quality Assessment of High Dynamic Range Sports Content”, IEEE/CVF Winter Conference on Applicaions of Computer Vision (WACV), 2023, pp. 556-564

[16] A. Telili, S. A. Fezza, W. Hamidouche, and H. F. Z. B. Meftah, ‘2BiVQA: Double Bi-LSTM Based Video Quality Assessment of UGC Videos’, ACM Trans. Multimedia Comput. Commun. Appl., Nov. 2023.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[18] J. Kufa, T. Kratochvil, “Comparison of H.265 and VP9 coding efficiency for full HDTV and ultra HDTV applications,” *Proceedings of 25th International Conference Radioelektronika, RADIOELEKTRONIKA 2015*, pp. 168–171, Jun. 2015, doi: 10.1109/RADIOELEK.2015.7128999.