VisualOn Optimizer Content Adaptive Encoding vs CBR and CRF

Executive Summary

The Channel Store (TCS), part of TVUp Group, is a leading provider of cloud-based playout and OTT distribution solutions, delivering linear and on-demand content for broadcasters across Europe and Latin America.

The Channel Store conducted an independent, detailed comparative analysis of CBR and CRF encoding versus VisualOn's Optimizer on a wide variety of content. Results demonstrated that Optimizer achieved the fastest encoding, lowest CPU demand, improved perceptual quality, and saved an average of 65% in bitrate over CBR.

Given its simple incorporation into existing workflows and the huge impact it has on the encoding results, Content-Adaptive Encoding (CAE) is the optimal choice for optimizing encoder efficiency.

Context

As global OTT video consumption continues to accelerate, broadcasters and OTT platforms face a growing challenge: delivering a high-quality viewing experience while controlling bandwidth and storage costs. The Channel Store (TCS), a subsidiary of TVUp, partnered with VisualOn to evaluate a new approach to improve video encoding efficiency without compromising visual fidelity.

Through a comprehensive comparative analysis of traditional Constant Bitrate (CBR), Constant Rate Factor (CRF), and VisualOn's AI-driven Optimizer, TCS identified significant quality and efficiency gains. Tests conducted across diverse content types—including sports, news, animation, and film—demonstrated that the VisualOn Optimizer achieved up to 40% bitrate savings while maintaining or even improving perceptual quality.

The collaboration highlights how innovative, AI-assisted encoding strategies can transform operational efficiency for OTT, helping them enhance viewer experience, reduce infrastructure load, and accelerate global scalability.

Market Context and Challenges

Streaming operators today face dual pressures: rising user expectations for high-definition, low-latency video, and escalating delivery costs due to higher resolutions and longer viewing times.

For operators like The Channel Store—serving audiences across Europe and Latin America—efficiency, agility, and compliance are key differentiators. Competing in a market alongside global cloud-based playout providers such as Amagi, The Channel Store focuses on regional content strategies, revenue-sharing monetization, and adherence to European broadcasting standards.

Against this backdrop, encoding optimization has become a strategic imperative. Traditional encoding methods (e.g., CBR or CRF) often struggle to balance video quality with data efficiency across diverse content types and network conditions. VisualOn Optimizer was evaluated as a next-generation solution designed to address this trade-off through AI-based perceptual optimization.

Collaboration Overview: VisualOn and The Channel Store

In collaboration with VisualOn, TCS conducted a series of in-depth encoding tests to assess the efficiency, stability, and visual quality of the VisualOn Optimizer in real-world production environments.

VisualOn, a pioneer in multimedia playback and optimization technologies, developed the Optimizer to apply AI enhanced content-adaptive perceptual analysis across video frames. The goal was to achieve superior compression efficiency, dynamically adjusting encoding parameters to preserve visual quality while minimizing bandwidth and maintaining visual quality.

This joint evaluation enabled TCS to benchmark the VisualOn Optimizer against conventional encoding methods and assess its potential for operational deployment in VOD workflows.

Testing Methodology

The evaluation focused on VOD encoding workflows, with outputs encoded using AVC for video and AAC for audio. No bitrate limits were imposed during testing, allowing the assessment of each method's full potential under unconstrained conditions.

TCS used the standard VisualOn Optimizer plugin for FFmpeg (based on FFmpeg, libx264) and after compilation could easily test Optimizer in their current transcoding workflow. See Annex A for the command lines used.

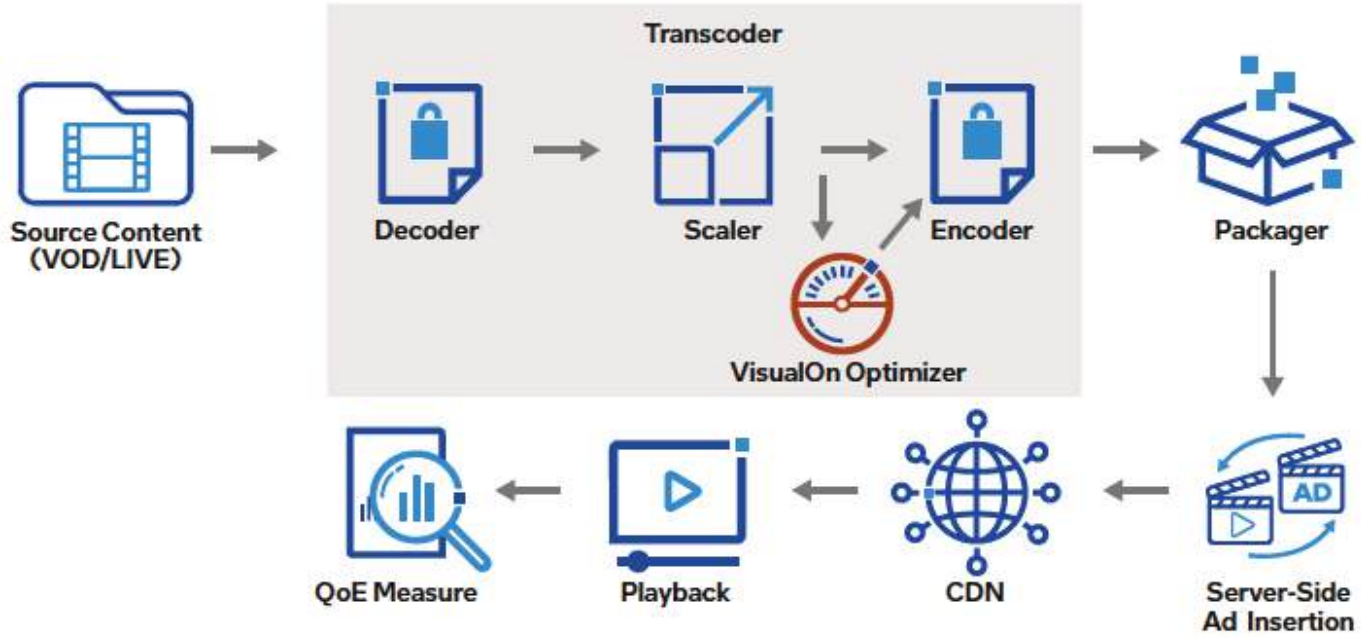


Figure 1. VisualOn Optimizer integrates into the streaming workflow

Encoding Methods Compared:

- Constant Bitrate (CBR)
- Constant Rate Factor (CRF)
- VisualOn Optimizer

Test Content Categories:

- Film / Drama
- Sports (high motion)
- News / Talk shows
- Animation

Each content type was analyzed for bitrate efficiency, visual quality (objective and subjective), and overall compression performance.

TCS noted that applying a fixed bitrate limit could yield further bitrate savings in high-motion sequences but might introduce minimal perceptual quality trade-offs—an important insight for real-world deployment optimization.

Comparative Analysis: CBR vs. CRF vs. VisualOn Optimizer

Constant Bitrate (CBR)

CBR encoding maintains a fixed bitrate throughout the video, ensuring predictable file sizes and consistent network load. However, this approach often leads to inefficiencies—allocating too many bits to low-motion scenes and too few to complex, high-motion segments—resulting in uneven visual quality and wasted bandwidth.

Constant Rate Factor (CRF)

CRF dynamically adjusts bitrate based on scene complexity, providing more efficient compression than CBR. Yet, while CRF improves quality consistency, it lacks fine-grained perceptual optimization. The encoded output may still exhibit visual degradation in challenging motion sequences or areas of high texture detail.

It operates by targeting a specific quality level rather than a fixed bitrate. This value defines the desired visual quality, within a range from lower values (higher quality and higher bitrate) to higher values (lower quality and lower bitrate). As a result, bitrate savings directly depend on the selected CRF target and the complexity of the content being encoded.

VisualOn Optimizer (VBR)

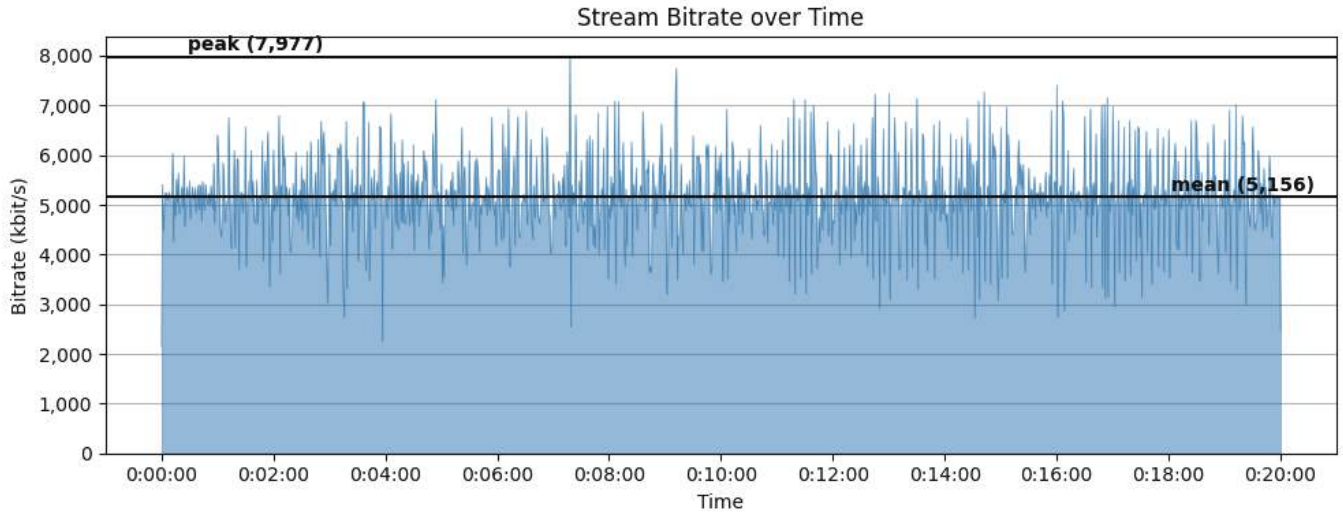
VisualOn's Optimizer employs AI-driven perceptual modeling and content-adaptive analysis to determine optimal encoding parameters per frame. This enables finer control over compression efficiency, allocating bits intelligently based on scene motion, brightness, and texture complexity.

VisualOn Optimizer is enabled by compiling FFmpeg with the Optimizer and adding arguments in the command line. See Annex A for the command lines used in this benchmark.

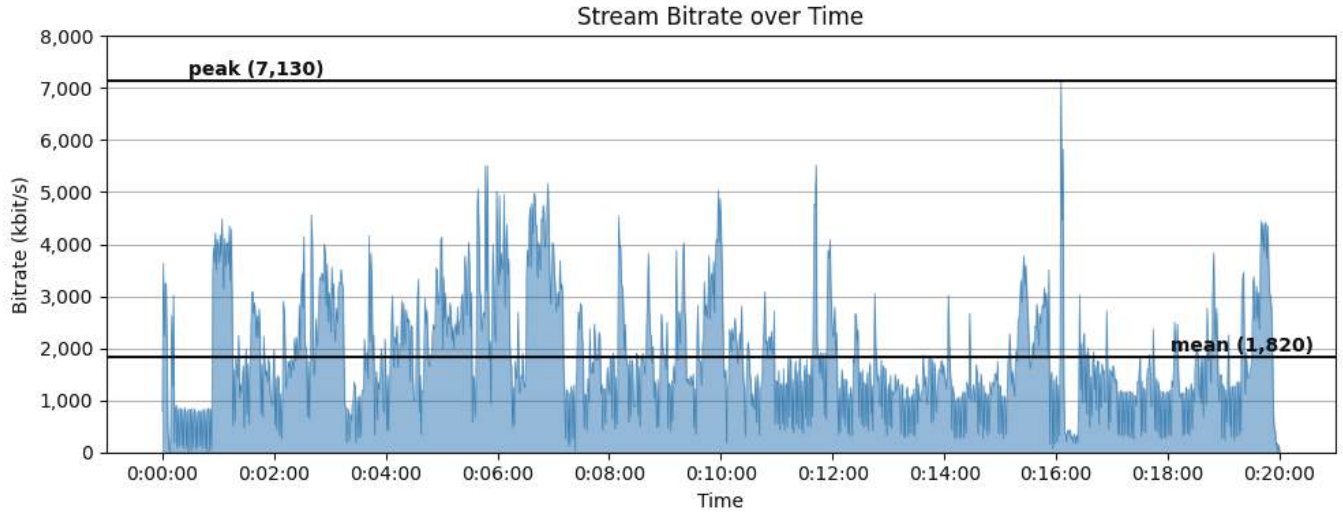
1. Sports Content: High-Motion Scenarios

Content Type	Content	Resolution	CBR (kbps)	Optimizer VBR (Kbps)	VMAF VO	CRF 21 (Kbps)	VMAF CRF	CRF vs. CBR (%)		Optimizer vs. CBR (%)		Optimizer vs. CRF (%)	
Sports	MMA	1080p	5160	4372	94.28	4602	90.48	10.81 %	17.92 %	15.27 %	45.10 %	5.00 %	34.36 %
		720p	2850	2560		2512		11.86 %		10.18 %		-1.91 %	
		480p	1860	1585		1657		10.91 %		14.78 %		4.35 %	
	Padel	1080p	5160	1820	96.76	4473	95.95	13.31 %		64.73 %		59.31 %	
		720p	2850	909		2280		20.00 %		68.11 %		60.13 %	
		480p	1860	572		1291		30.59 %		69.25 %		55.69 %	
	Surf	1080p	5160	2490	95.68	4368	94.34	15.35 %		51.74 %		42.99 %	
		720p	2850	1333		2289		19.68 %		53.23 %		41.76 %	
		480p	1860	770		1325		28.76 %		58.60 %		41.89 %	

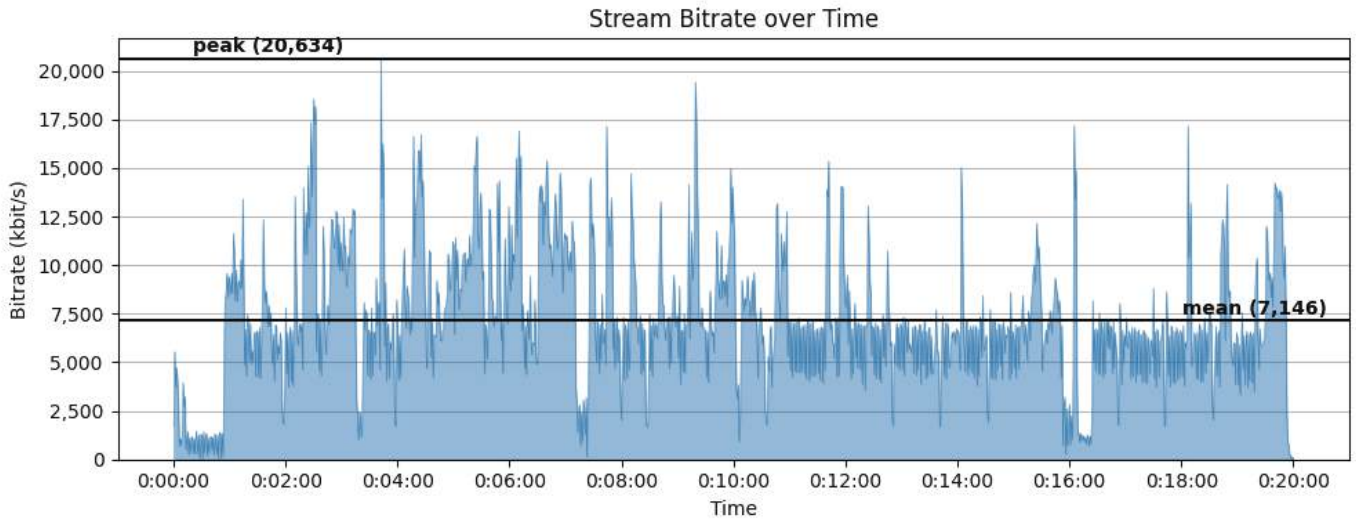
Table1. Average Bitrate Savings by Encoding Method Across Sports Content



Bitrate - Current Encoding

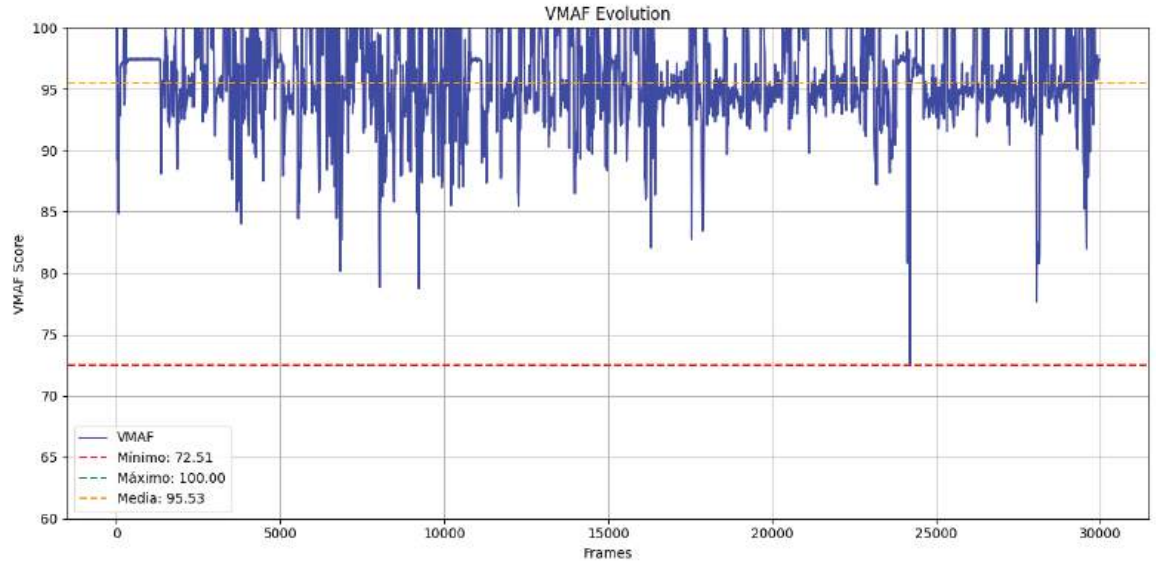


Bitrate - VisualOn Optimizer Encoding

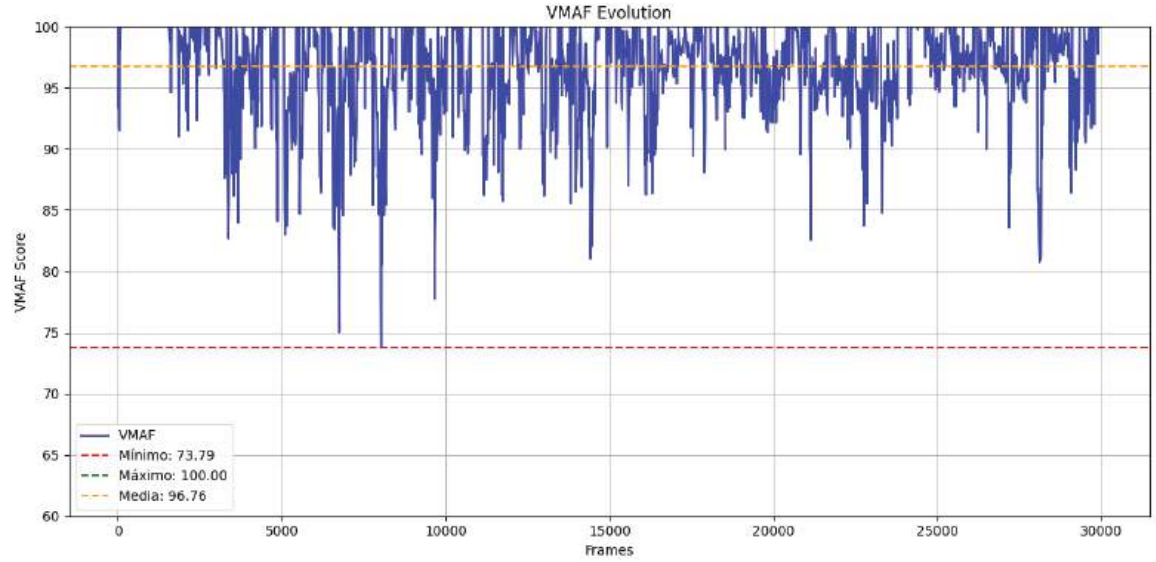


Bitrate - CRF Encoding

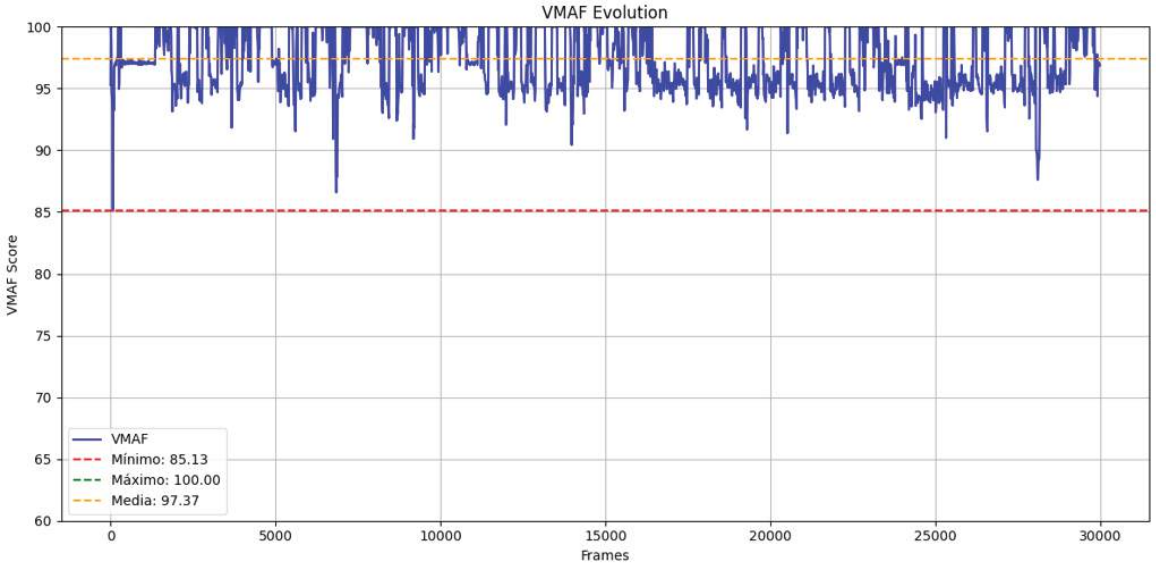
Figure 1-1. Bitrate comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *My Padel TV: Hexagon Cup 2024 Highlights Dia 4*



VMAF - Current Encoding



VMAF - VisualOn Optimizer Encoding



VMAF - CRF Encoding

Figure 1-2. VMAF quality comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *My Padel TV: Hexagon Cup 2024 Highlights Dia 4*

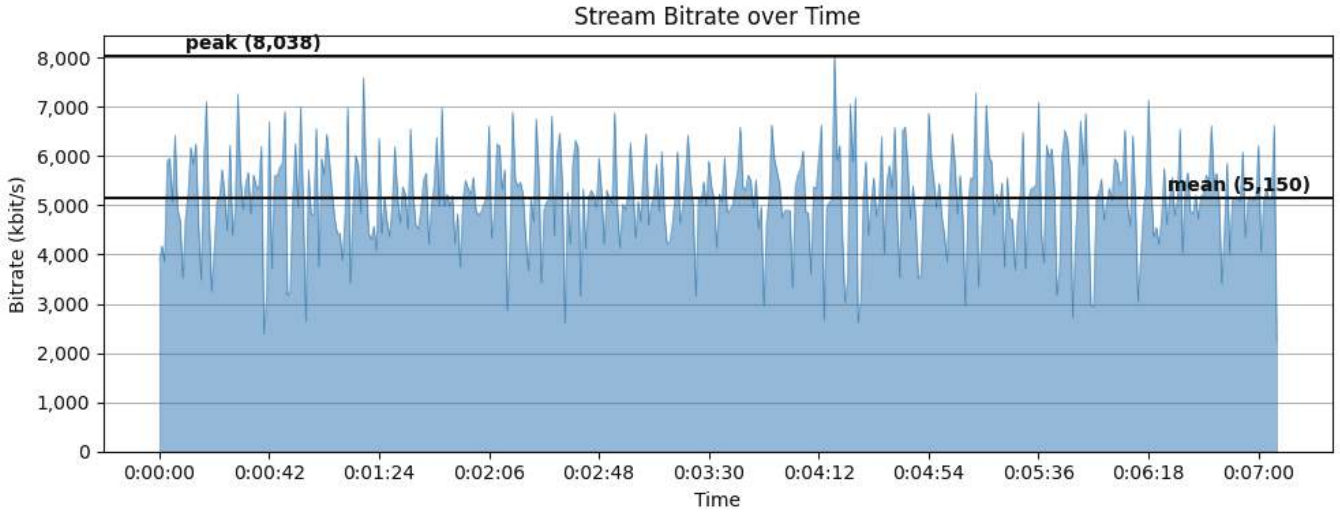
Sports footage achieved significant bitrate reductions of 45% on average despite its demanding compression characteristics. Savings ranged from 13% (MMA) to 67% (padel) and 55% (surf).

The Optimizer maintained or improved VMAF scores, demonstrating excellent adaptability to rapid motion and dynamic scene transitions.

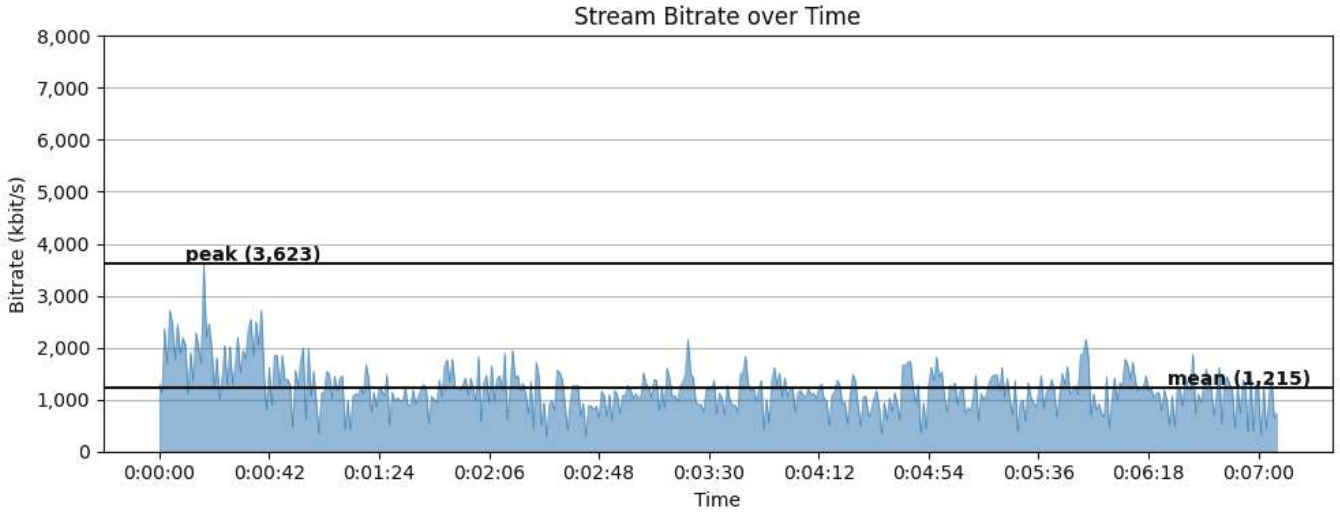
2. News and Documentaries: Efficiency in Controlled Environments

Content Type	Content	Resolution	CBR (Kbps)	Optimizer VBR (Kbps)	VMAF VO	CRF 21 (Kbps)	VMAF CRF	CRF vs. CBR (%)		Optimizer vs. CBR (%)		Optimizer vs. CRF (%)	
News / Documentary	Curry (El País)	1080p	5160	1215	95.75	4316	95.27	16.36 %	25.29 %	76.45 %	68.85 %	71.85 %	58.23 %
		720p	2850	671		1991		30.14 %		76.46 %		66.30 %	
		480p	1860	411		1027		44.78 %		77.90 %		59.98 %	
	Hora Veintipico (El país)	1080p	5160	1504	96.83	4616	94.88	10.54 %		70.85 %		67.42 %	
		720p	2850	680		2004		29.68 %		76.14 %		66.07 %	
		480p	1860	440		1021		45.11 %		76.34 %		56.90 %	
	Obama (Spica Life)	1080p	5160	2214	95.00	4652	95.15	9.84 %		57.09 %		52.41 %	
		720p	2850	1093		2412		15.37 %		61.65 %		54.68 %	
		480p	1860	647		1366		26.56 %		65.22 %		52.64 %	
	Chirurgie (Spica Life)	1080p	5160	2308	94.85	4615	94.37	10.56 %		55.27 %		49.99 %	
		720p	2850	1051		2156		24.35 %		63.12 %		51.25 %	
		480p	1860	564		1113		40.16 %		69.68 %		49.33 %	

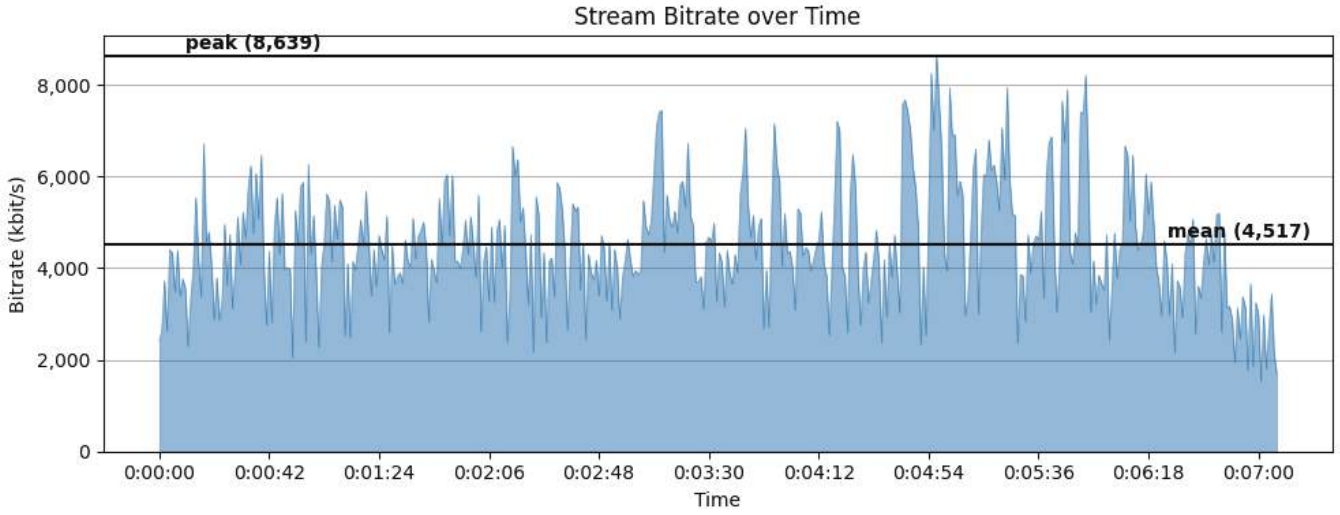
Table2. Average Bitrate Savings by Encoding Method Across News/Documentary Content



Bitrate - Current Encoding

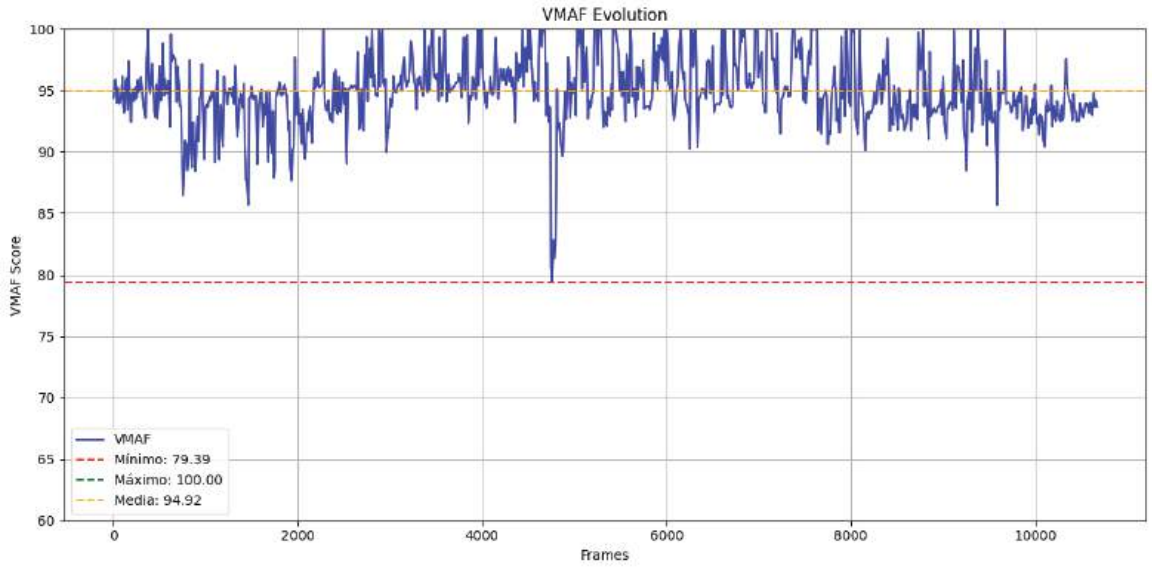


Bitrate - VisualOn Optimizer Encoding

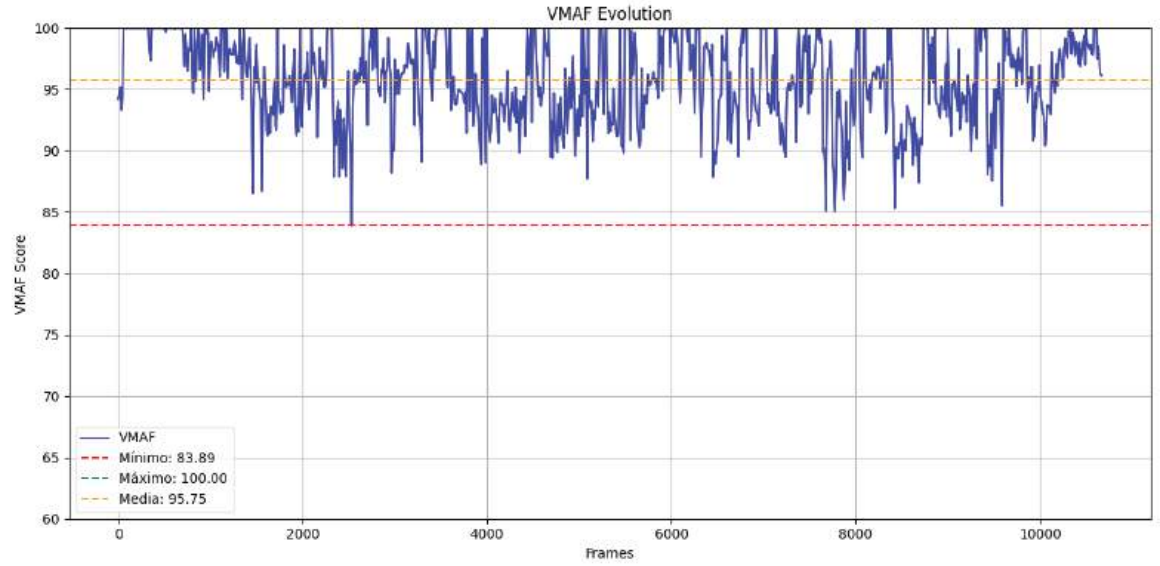


Bitrate - CRF Encoding

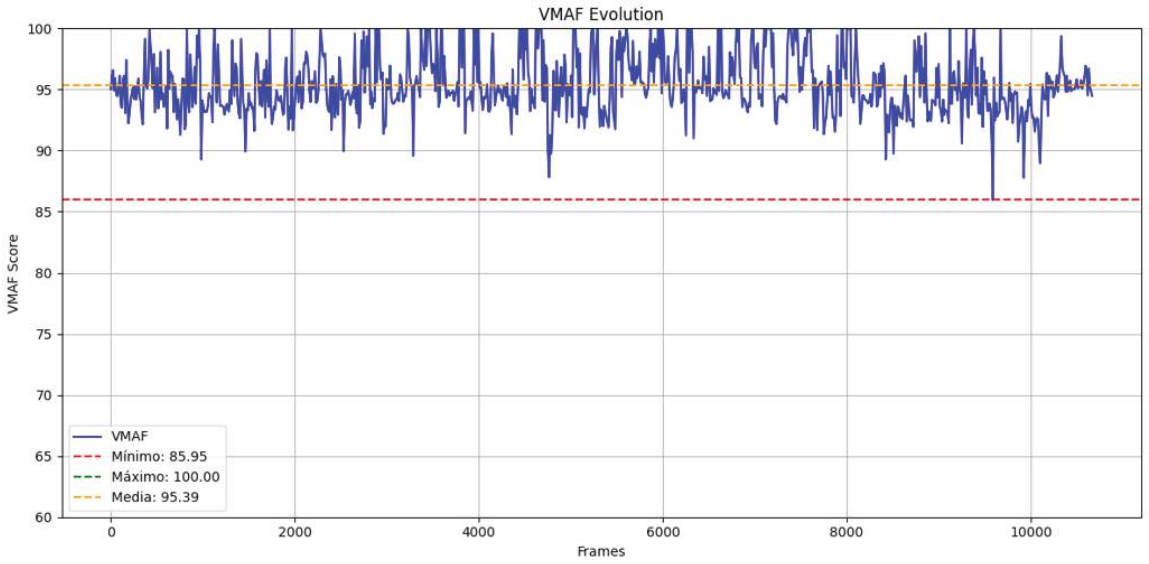
Figure 2-1. Bitrate comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *El País: Curry japonés, una receta de primero de cocina*



VMAF - Current Encoding



VMAF - VisualOn Optimizer Encoding



VMAF - CRF Encoding

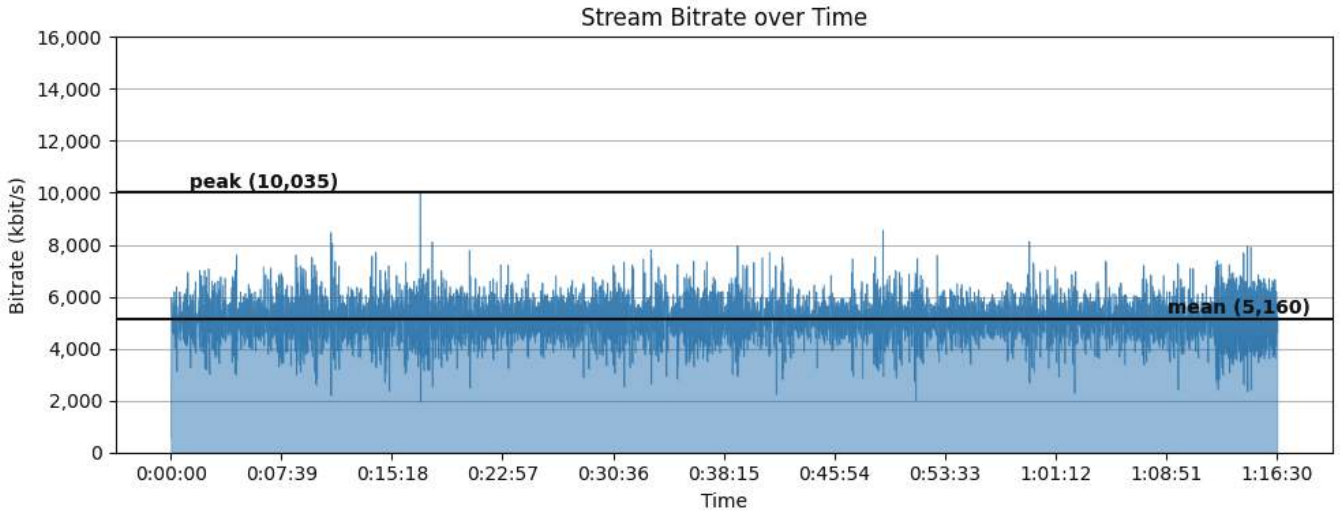
Figure 2-2. VMAF quality comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *El País: Curry japonés, una receta de primero de cocina*

Characterized by moderate motion and studio-based production, this category achieved 68.8% average bitrate savings across all resolutions. The Optimizer efficiently managed added visual elements such as graphics, subtitles, and logos, maintaining or enhancing overall visual quality.

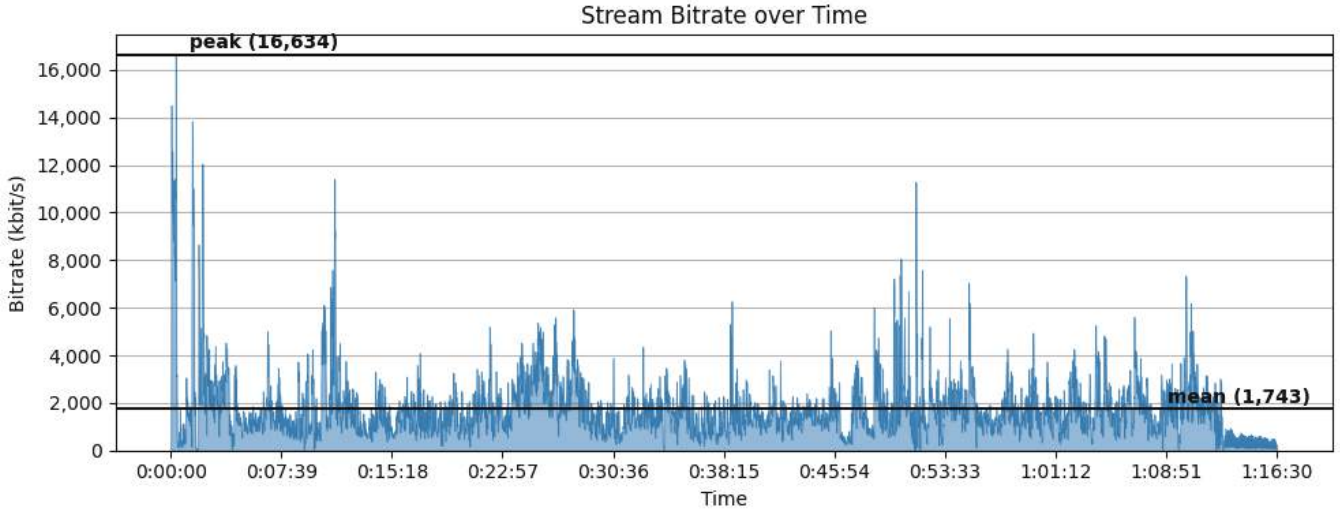
3. Film and Series: Consistent Quality in Long-Form Content

Content Type	Content	Resolution	CBR (Kbps)	Optimizer VBR (Kbps)	VMAF VO	CRF 21 (Kbps)	VMAF CRF	CRF vs. CBR (%)		Optimizer vs. CBR (%)		Optimizer vs. CRF (%)	
Cinema / Film	Ana de las tejas verdes (Anne of Green Gables)	1080p	5160	2283	94.51	3732	90.96	27.67 %	39.57 %	55.76 %	64.84 %	38.83 %	40.05 %
		720p	2850	920		1520		46.67 %		67.72 %		39.47 %	
		480p	1860	456		726		60.97 %		75.48 %		37.19 %	
	El callejón (The Alley)	1080p	5160	1743	93.98	3358	91.82	34.92 %		66.22 %		48.09 %	
		720p	2850	818		1227		56.95 %		71.30 %		33.33 %	
		480p	1860	445		593		68.12 %		76.08 %		24.96 %	
	Junimond (Junimond)	1080p	5160	2817		4978		3.53 %		45.41 %		43.41 %	
		720p	2850	1234		2417		15.19 %		56.70 %		48.94 %	
		480p	1860	579		1076		42.15 %		68.87 %		46.19 %	

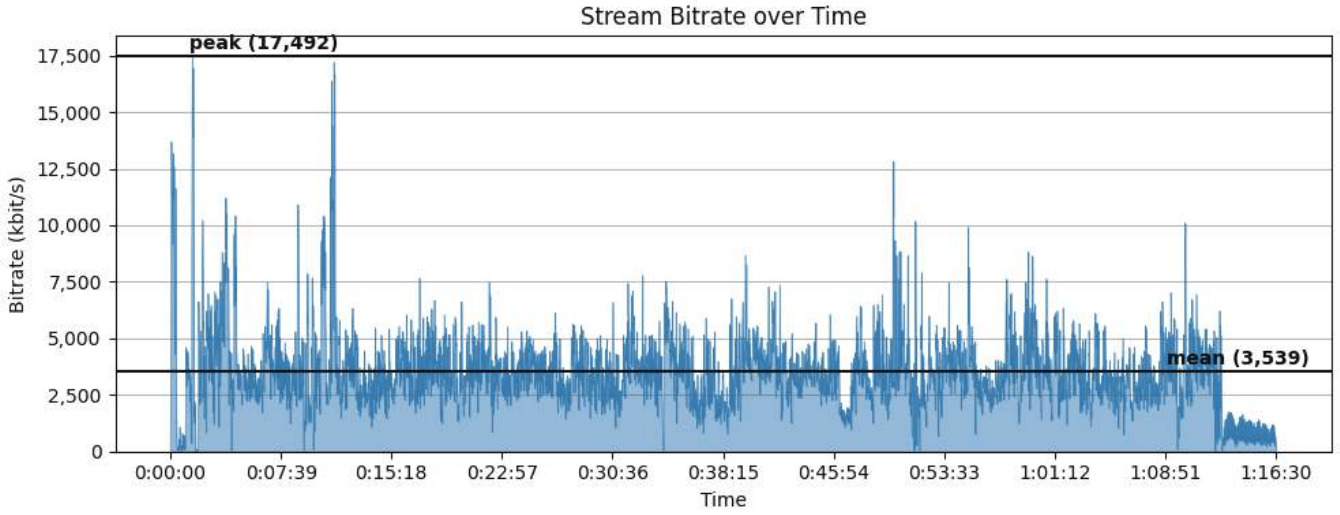
Table3. Average Bitrate Savings by Encoding Method Across Film and Series Content



Bitrate - Current Encoding

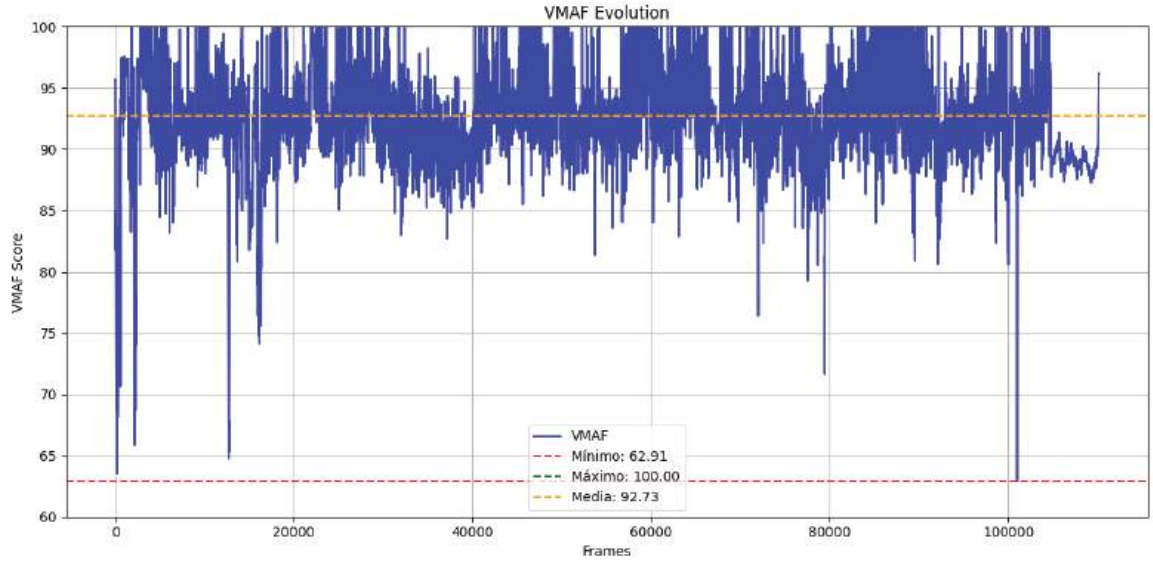


Bitrate - VisualOn Optimizer Encoding

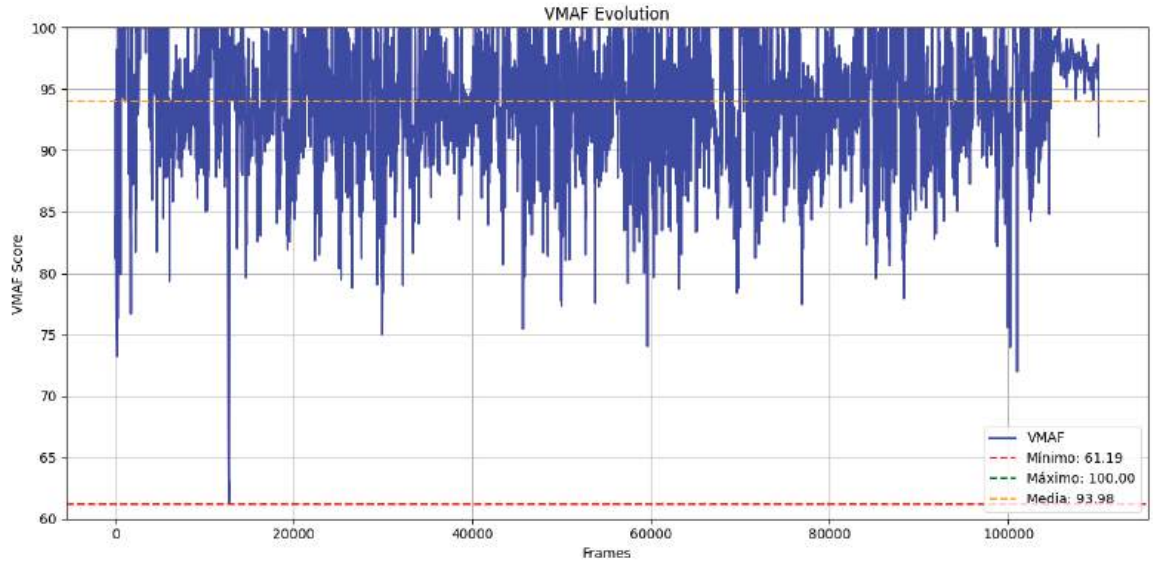


Bitrate - CRF Encoding

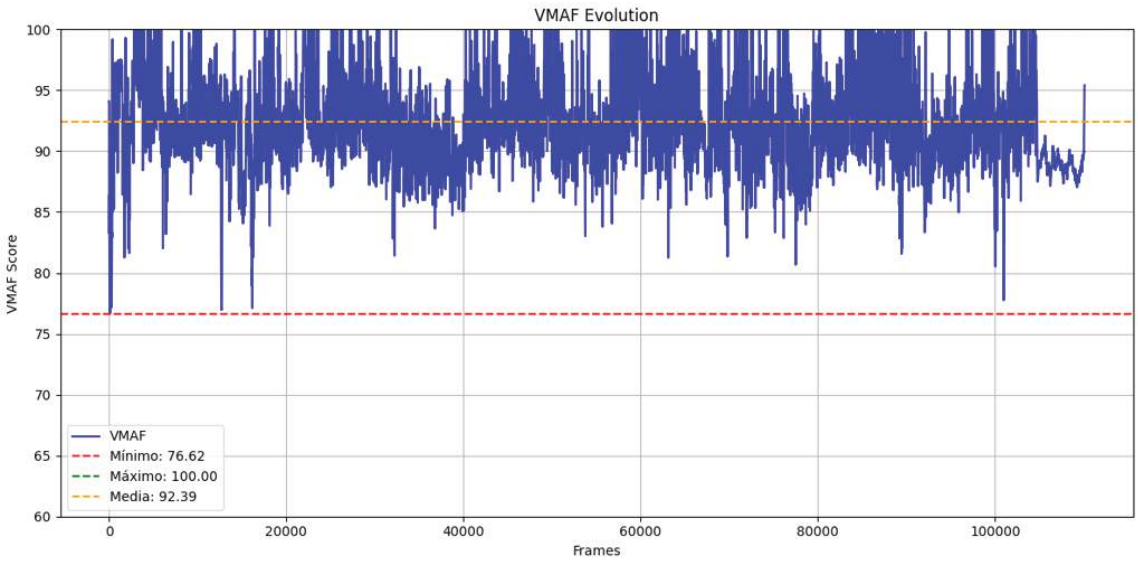
Figure 3-1. Bitrate comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *Planeta de Cine: El Callejón*.



VMAF - Current Encoding



VMAF - VisualOn Optimizer Encoding



VMAF - CRF Encoding

Figure 3-2. VMAF quality comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *Planeta de Cine: El Callejón*.

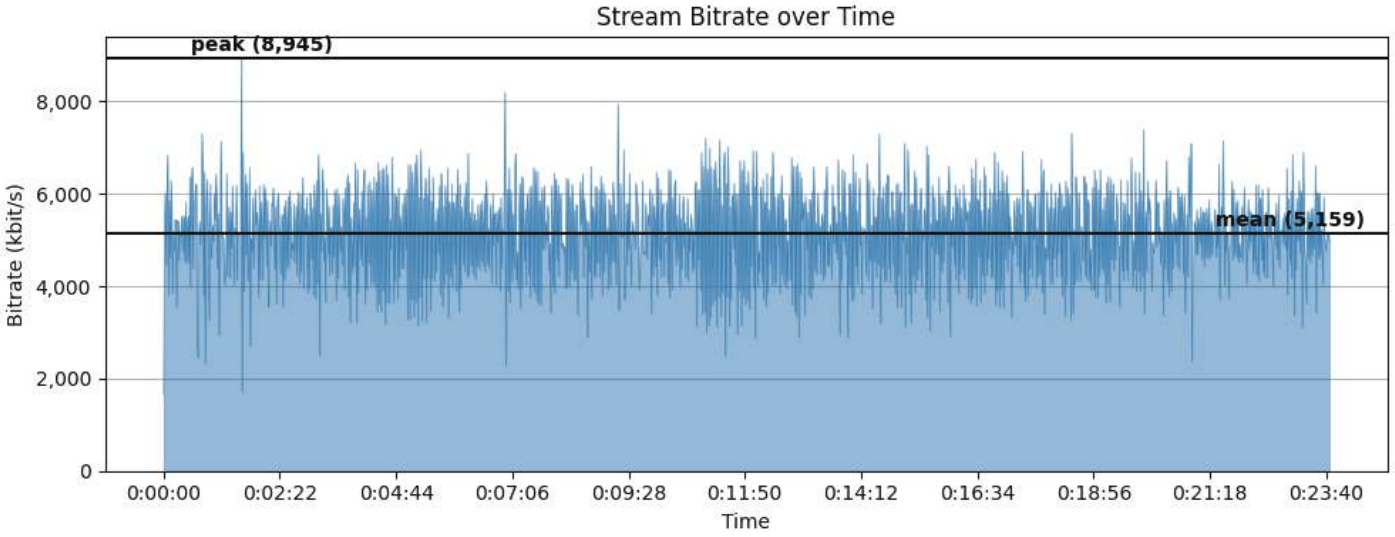
Narrative content featuring a mix of static and dynamic scenes achieved consistent bitrate savings without compromising visual quality. Average reduction reached 64.8%, with low-motion sequences contributing significantly to efficiency.

Compared to CRF, maintaining equivalent VMAF scores would require ~80% higher bitrate, whereas the Optimizer reduced bitrate by ~20% versus current encoding profiles at the same quality level.

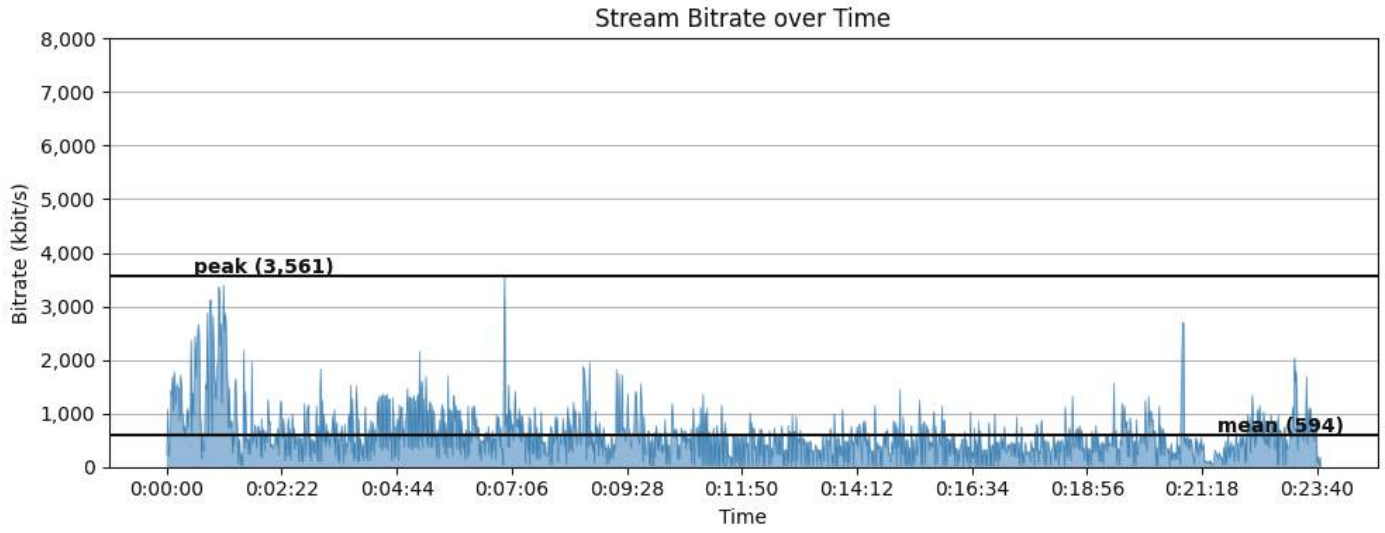
4. Animation: Exceptional Gains in Stylized Content

Content Type	Content	Resolution	CBR (Kbps)	Optimizer VBR (Kbps)	VMAF VO	CRF 21 (kbps)	VMAF CRF	CRF vs. CBR (%)		Optimizer vs. CBR (%)		Optimizer vs. CRF (%)	
Animation	Arifureta	1080p	5160	594	96.18	2521	94.75	51.14 %	56.32 %	88.49 %	85.81 %	76.44 %	65.73 %
		720p	2850	332		977		65.72 %		88.35 %		66.02 %	
		480p	1860	226		485		73.92 %		87.85 %		53.40 %	
	Megalobox	1080p	5160	625	95.45	2657	94.29	48.51 %		87.89 %		76.48 %	
		720p	2850	300		1012		64.49 %		89.47 %		70.36 %	
		480p	1860	208		487		73.82 %		88.82 %		57.29 %	
	Inuyasha	1080p	5160	951	95.19	3721	93.88	27.89 %		81.57 %		74.44 %	
		720p	2850	555		1578		44.63 %		80.53 %		64.83 %	
		480p	1860	384		805		56.72 %		79.35 %		52.30 %	

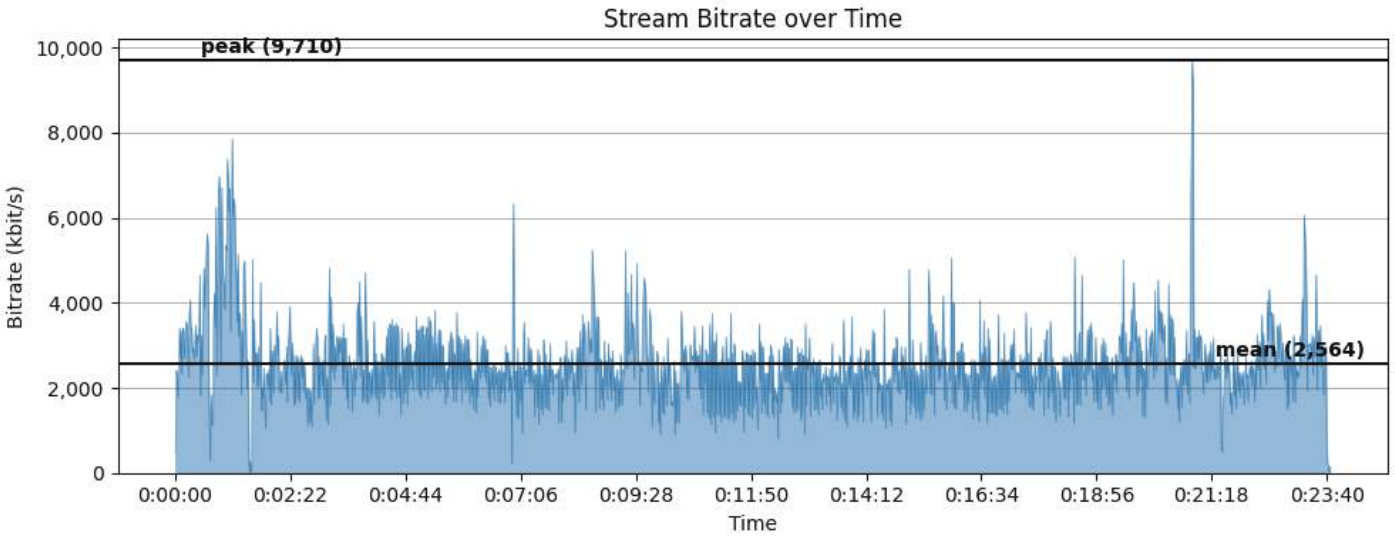
Table4. Average Bitrate Savings by Encoding Method Across Animation Content



Bitrate - Current Encoding

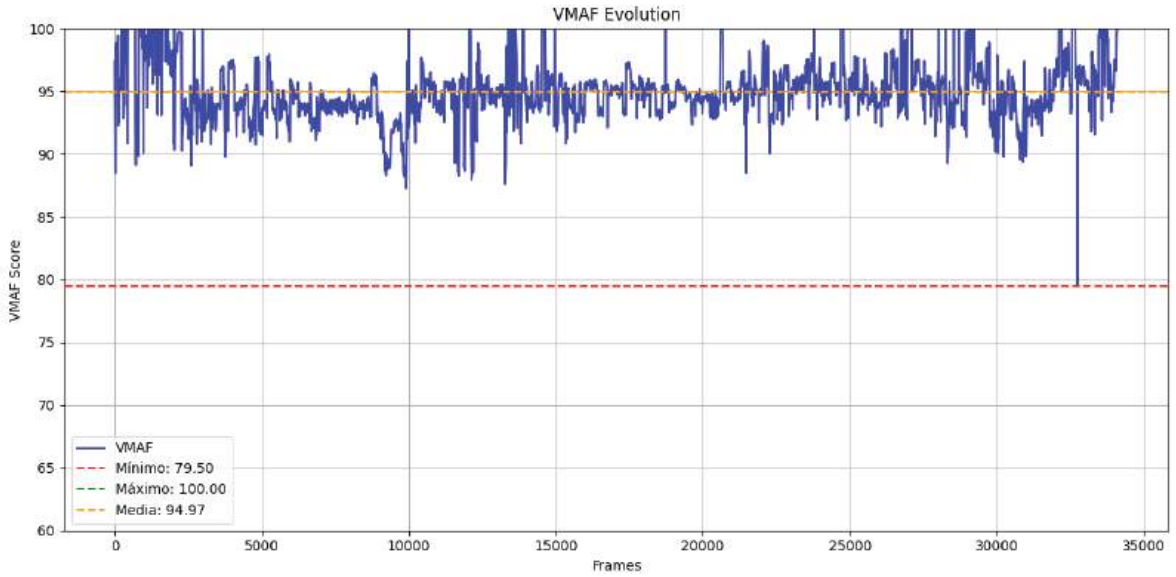


Bitrate - VisualOn Optimizer Encoding

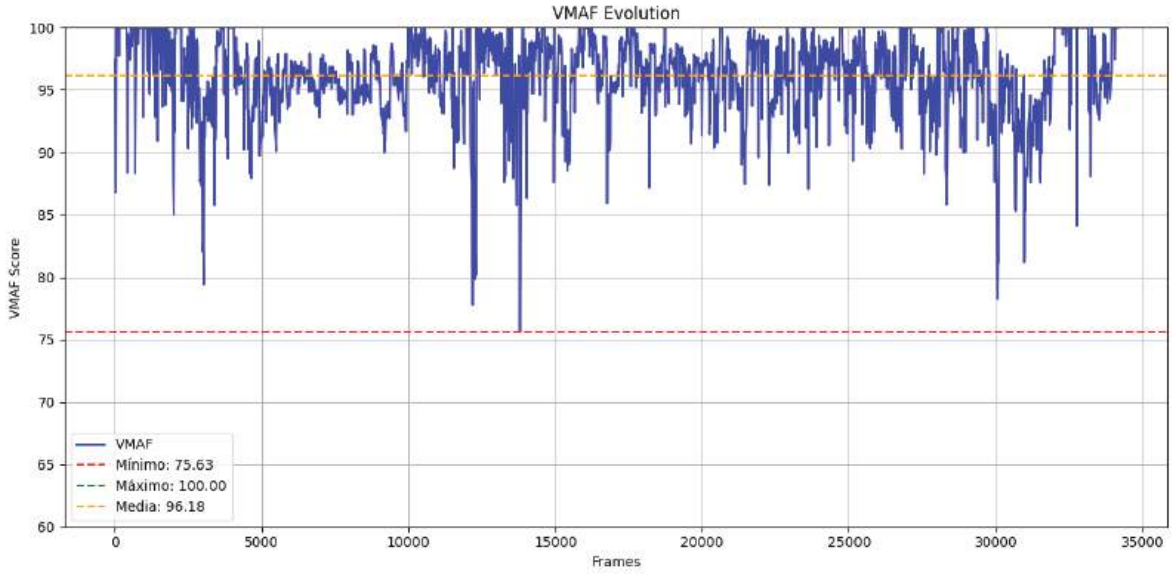


Bitrate - CRF Encoding

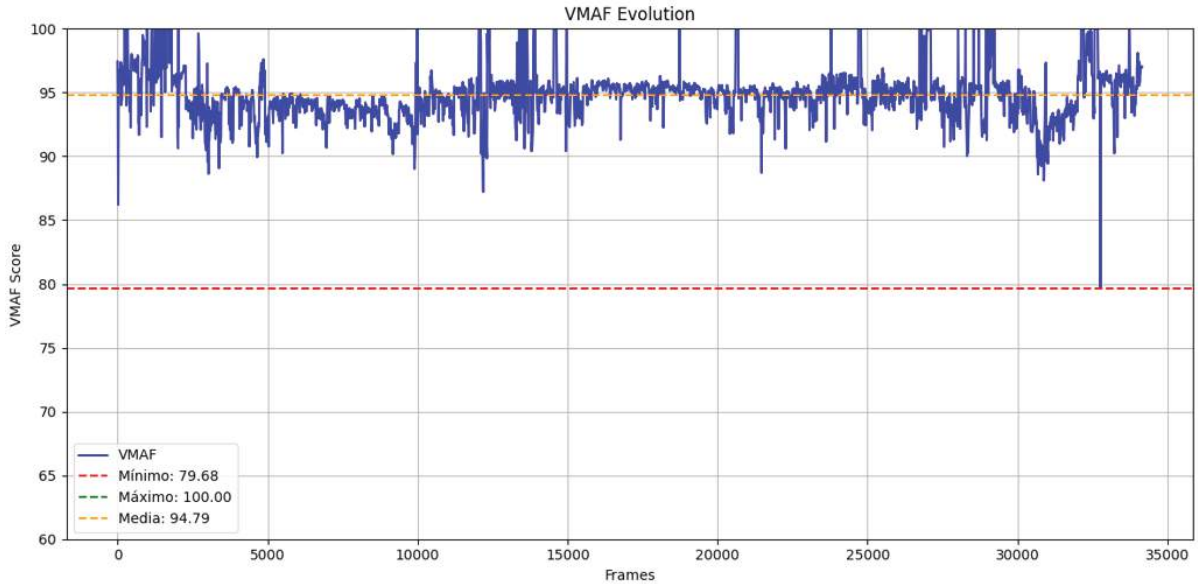
Figure 4-1. Bitrate comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *Anime Visión: Arifureta T02 E12, Un nuevo viaje*



VMAF - Current Encoding



VMAF - VisualOn Optimizer Encoding



VMAF - CRF Encoding

Figure 4-2. VMAF quality comparison between Current Encoding, VisualOn Optimizer Encoding, and CRF Encoding on the source video *Anime Visión: Arifureta T02 E12, Un nuevo viaje.*

Animated content delivered the highest bitrate savings of 85.8% on average thanks to large flat-color regions, sharp edges, and minimal noise. While CRF achieved 26–52% savings compared to CBR, it required 300–320% more bitrate than VBR, underscoring the Optimizer's exceptional efficiency for highly stylized and less complex visuals.

Analysis of Peak Bitrates – Why and How to Tackle

While the overall bitrate reduction achieved by Optimizer is substantial, identifying scenarios where compression is less effective is equally important. To this end, peak bitrate segments across all VBR-encoded outputs were analyzed.

This analysis aims to detect scenes with the highest instantaneous bitrate after optimization and identify encoder challenges or limitations that the encoding with Optimizer may face.

The highest bitrate peaks were consistently observed in scenes with special lighting conditions, such as light bulbs turning on in dark environments. In general, dark scenes require more bitrate to maintain visual quality, likely due to noise and fine gradients in shadowed areas that are harder to compress effectively.

Other common peak triggers include:

- Scenes with water, especially moving water with reflections or splashes as seen in the lake scene or the surf channel
- Grass fields or foliage, due to their fine texture and irregular patterns
- Snowy environments, which often introduce a mix of brightness and subtle detail
- Wide landscapes in general, which combine high detail, depth, and complex motion across large areas

These patterns suggest that the encoder and thus Optimizer require more bits to achieve high quality with high spatial complexity and variable lighting.

Should wide bitrate variation be a concern, it can be controlled by setting the appropriate maxrate and bufsize parameters with the encoder. This can achieve a slight additional bitrate saving high-motion scenes with bitrate spikes at the cost of a minor reduction in quality in those moments.

Encoding Performance Analysis

In addition to the quality and bitrate comparisons, the encoding performance of the three methods (CBR, CRF, and Optimizer) was also evaluated. The performance assessment was conducted locally on a laptop, using one representative sample for each content category. Although the absolute encoding times are specific to this environment, the relative percentages reported here reliably reflect the tendencies of each method and can be considered representative of their general behavior.

The metrics analyzed included total encoding time, CPU usage, and memory consumption. Some clear tendencies emerged across all test cases:

- **Encoding Time (rtime):**
  - CRF reduced encoding time by 3–25% vs. CBR (avg. 19%).
  - Optimizer achieved 7–27% faster encoding (avg. 22%).
  - Optimizer remained ~2–4% faster than CRF.
- **CPU Usage (utime):**
  - CRF lowered CPU use by 15% vs. CBR, up to 31% in some cases.
  - Optimizer further reduced CPU consumption by up to 50%, averaging 35% less.
  - Optimizer offered 21–28% additional savings above CRF.
- **Memory Usage (maxrss):**
  - CRF slightly reduced memory (up to 2%) vs. CBR.
  - Optimizer required 16–32% more memory than CBR and 19–35% more than CRF.

In summary, CRF offers a balanced improvement over CBR in both speed and memory efficiency, whereas VisualOn Optimizer achieves the fastest encoding and lowest CPU demand at the expense of more memory consumption.

Final Conclusion: Bitrate Savings and Quality Impact

Comprehensive testing of the VisualOn Optimizer across a diverse range of video content—including animation, sports, film and series, and news/documentaries—revealed consistent patterns in bitrate reduction and quality retention.

Global Bitrate Savings

For each category, the average bitrate savings across all three video layers (1080p, 720p, 480p) were:

Category	Average Bitrate Savings (VisualOn Optimizer)	Average Bitrate Savings (CRF)
Film & Series	64.84%	39.5%
Sports	45.10%	17.9%
News & Documentaries	68.8%	25.3%
Animation	85.8%	56.3%

These savings are calculated by comparing the original CBR-encoded content (5160/2850/1860 kbps for 1080p/720p/480p, respectively) with the newly generated Optimizer VBR-encoded versions.

When aggregating results across all categories (weighted equally per content and layer) and without considering additional layers some content may include, the estimated overall bitrate savings is approximately **65%** when applying the Optimizer at scale across the full catalog.

This represents a significant improvement in storage and delivery efficiency, reducing costs for both CDN and storage infrastructure.

CRF vs. CBR vs. VisualOn Optimizer (VBR)

CRF clearly improves upon current CBR encoding and results vary with content type. In some cases, bitrate reductions are smaller while others exceed 50%, with an average of 20%. Compared to VBR with VisualOn Optimizer, CRF typically requires significantly more bitrate, ranging from about 80% up to over 300% in certain cases.

The primary advantage of CRF is its ability to maintain consistent and predictable quality by adapting bitrate to scene complexity. VBR with VisualOn Optimizer, in contrast, optimizes for lower overall bitrate but allows more variability in visual quality. The estimated overall bitrate savings when applying CRF encoding across the full catalog is approximately 34%.

In this study, CRF encoding was performed using a fixed target value of CRF 21 across all tested content. This value was selected as a best-fit average, as it resulted in an overall VMAF score closest to the target quality level used for the VisualOn Optimizer (VMAF 96).

It should be noted that CRF evaluations involve a degree of approximation. Depending on content complexity, higher CRF values could in some cases still achieve the same target VMAF, leading to additional bitrate savings.

Therefore, while the results presented here provide a representative and fair comparison, further per-title tuning of CRF values could potentially achieve even closer quality matching or incremental efficiency gains.

Optimizer Visual Quality (VMAF) Results

In addition to bandwidth savings, VMAF scores were used to assess perceived visual quality. Across all tested content, VMAF scores remained stable or improved by 1–3 points after optimization.

These results confirm that the Optimizer not only compresses more efficiently but also enhances visual quality.

Summary

The collaboration between VisualOn and The Channel Store demonstrates the transformative potential of AI-driven video optimization. Across diverse content types—sports, news, film, and animation—the VisualOn Optimizer consistently delivered substantial bitrate savings, averaging approximately 65% across all resolutions, while maintaining or enhancing perceived visual quality (VMAF).

Compared to traditional CBR and CRF encoding, Optimizer achieved faster encoding times and significantly lower CPU usage, albeit with higher memory demand. These results highlight how AI-assisted, content-adaptive encoding can reduce delivery costs, improve operational efficiency, and ensure superior viewer experiences for OTT platforms at scale, without any need to proceed to per-title or manual encoding adjustments : it is all done automatically by Optimizer.

Annex A – Command Lines

=== CBR ENCODING ===

```
ffmpeg -hide_banner -benchmark -nostdin -y -xerror \
-i “promoTivify1.mp4” -max_muxing_queue_size 4096 \
-filter_complex “\
[0:v:0]scale=iw*sar:ih,yadif[v_step_0]; \
[v_step_0]null[v_step_1]; \
[v_step_1]format=yuv420p,split=5[v_step_2_0][v_step_2_1][v_step_2_2][v_step_2_3][v_step_2_4]; \
[v_step_2_0]scale=1920:1080:force_original_aspect_ratio=decrease,pad=1920:1080:-1:-
1:color=black[v_out_0]; \
[v_step_2_1]scale=1280:720:force_original_aspect_ratio=decrease,pad=1280:720:-1:-
1:color=black[v_out_1]; \
[v_step_2_2]scale=854:480:force_original_aspect_ratio=decrease,pad=854:480:-1:-1:color=black[v_out_2];
\
[v_step_2_3]scale=640:360:force_original_aspect_ratio=decrease,pad=640:360:-1:-1:color=black[v_out_3];
\
[v_step_2_4]scale=426:240:force_original_aspect_ratio=decrease,pad=426:240:-1:-1:color=black[v_out_4];
\
[0:a:0]loudnorm=I=-23:LRA=10:TP=-1[a_step_0]; \
[a_step_0]asplit=5[a_out_0][a_out_1][a_out_2][a_out_3][a_out_4]” \
\
-map “[v_out_0]” -map “[a_out_0]” -c:v:0 libx264 \
-x264-params “nal-hrd=cbr:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -vsync cfr -r 25 -profile:v high -bf 2 \
-b:v:0 5161200 -maxrate:v:0 5161200 -minrate:v:0 5161200 -bufsize:v:0 10322400 \
-c:a:0 aac -b:a:0 128k -ac:0 2 -ar 48000 -metadata:s:a:0 language=spa \
\
-map “[v_out_1]” -map “[a_out_1]” -c:v:1 libx264 \
-x264-params “nal-hrd=cbr:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -vsync cfr -r 25 -profile:v high -bf 2 \
-b:v:1 2851200 -maxrate:v:1 2851200 -minrate:v:1 2851200 -bufsize:v:1 5702400 \
-c:a:1 aac -b:a:1 128k -ac:1 2 -ar 48000 -metadata:s:a:1 language=spa \
\
-map “[v_out_2]” -map “[a_out_2]” -c:v:2 libx264 \
-x264-params “nal-hrd=cbr:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -vsync cfr -r 25 -profile:v high -bf 2 \
-b:v:2 1861200 -maxrate:v:2 1861200 -minrate:v:2 1861200 -bufsize:v:2 3722400 \
-c:a:2 aac -b:a:2 128k -ac:2 2 -ar 48000 -metadata:s:a:2 language=spa \
\
-map “[v_out_3]” -map “[a_out_3]” -c:v:3 libx264 \
-x264-params “nal-hrd=cbr:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -vsync cfr -r 25 -profile:v high -bf 2 \
-b:v:3 926200 -maxrate:v:3 926200 -minrate:v:3 926200 -bufsize:v:3 1852400 \
-c:a:3 aac -b:a:3 128k -ac:3 2 -ar 48000 -metadata:s:a:3 language=spa \
\
```

```
-map “[v_out_4]” -map “[a_out_4]” -c:v:4 libx264 \
-x264-params “nal-hrd=cbr:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -vsync cfr -r 25 -profile:v high -bf 2 \
-b:v:4 486200 -maxrate:v:4 486200 -minrate:v:4 486200 -bufsize:v:4 972400 \
-c:a:4 aac -b:a:4 128k -ac:4 2 -ar 48000 -metadata:s:a:4 language=spa \
\
-f hls -hls_time 5 -hls_playlist_type vod \
-hls_flags independent_segments+discont_start+program_date_time \
-hls_segment_type mpegts \
-hls_segment_filename “prueba comandos/cbr/data_%v_%02d.ts” \
-master_pl_name master.m3u8 \
-var_stream_map “v:0,a:0 v:1,a:1 v:2,a:2 v:3,a:3 v:4,a:4” \
“prueba comandos/cbr/stream_%v.m3u8”
```

=== CRF ENCODING ===

```
ffmpeg -hide_banner -benchmark -nostdin -y -xerror \
-i “promoTivify1.mp4” -max_muxing_queue_size 4096 \
-filter_complex “\
[0:v:0]scale=iw*sar:ih,yadif[v_step_0]; \
[v_step_0]null[v_step_1]; \
[v_step_1]format=yuv420p,split=5[v_step_2_0][v_step_2_1][v_step_2_2][v_step_2_3][v_step_2_4]; \
[v_step_2_0]scale=1920:1080:force_original_aspect_ratio=decrease,pad=1920:1080:-1:-
1:color=black[v_out_0]; \
[v_step_2_1]scale=1280:720:force_original_aspect_ratio=decrease,pad=1280:720:-1:-
1:color=black[v_out_1]; \
[v_step_2_2]scale=854:480:force_original_aspect_ratio=decrease,pad=854:480:-1:-1:color=black[v_out_2];
\
[v_step_2_3]scale=640:360:force_original_aspect_ratio=decrease,pad=640:360:-1:-1:color=black[v_out_3];
\
[v_step_2_4]scale=426:240:force_original_aspect_ratio=decrease,pad=426:240:-1:-1:color=black[v_out_4];
\
[0:a:0]loudnorm=I=-23:LRA=10:TP=-1[a_step_0]; \
[a_step_0]asplit=5[a_out_0][a_out_1][a_out_2][a_out_3][a_out_4]” \
\
-map “[v_out_0]” -map “[a_out_0]” -c:v:0 libx264 -crf:v:0 21 -preset slow \
-x264-params “force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -r 25 -profile:v high -bf 2 \
-maxrate:v:0 5161200 -bufsize:v:0 10322400 \
-c:a:0 aac -b:a:0 128k -ac:0 2 -ar 48000 -metadata:s:a:0 language=spa \
\
-map “[v_out_1]” -map “[a_out_1]” -c:v:1 libx264 -crf:v:1 21 -preset slow \
-x264-params “force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -r 25 -profile:v high -bf 2 \
-maxrate:v:1 2851200 -bufsize:v:1 5702400 \
-c:a:1 aac -b:a:1 128k -ac:1 2 -ar 48000 -metadata:s:a:1 language=spa \
\
```

```
-map “[v_out_2]” -map “[a_out_2]” -c:v:2 libx264 -crf:v:2 21 -preset slow \
-x264-params “force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -r 25 -profile:v high -bf 2 \
-maxrate:v:2 1861200 -bufsize:v:2 3722400 \
-c:a:2 aac -b:a:2 128k -ac:2 2 -ar 48000 -metadata:s:a:2 language=spa \
\
-map “[v_out_3]” -map “[a_out_3]” -c:v:3 libx264 -crf:v:3 21 -preset slow \
-x264-params “force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -r 25 -profile:v high -bf 2 \
-maxrate:v:3 926200 -bufsize:v:3 1852400 \
-c:a:3 aac -b:a:3 128k -ac:3 2 -ar 48000 -metadata:s:a:3 language=spa \
\
-map “[v_out_4]” -map “[a_out_4]” -c:v:4 libx264 -crf:v:4 21 -preset slow \
-x264-params “force-cfr=1:scenecut=-1:min-keyint=50:keyint=50” \
-force_key_frames “expr:gte(t,n_forced*5.0)” -r 25 -profile:v high -bf 2 \
-maxrate:v:4 486200 -bufsize:v:4 972400 \
-c:a:4 aac -b:a:4 128k -ac:4 2 -ar 48000 -metadata:s:a:4 language=spa \
\
-f hls -hls_time 5 -hls_playlist_type vod \
-hls_flags independent_segments+discont_start+program_date_time \
-hls_segment_type mpegts \
-hls_segment_filename “prueba comandos/crf/data_%v_%02d.ts” \
-master_pl_name master.m3u8 \
-var_stream_map “v:0,a:0 v:1,a:1 v:2,a:2 v:3,a:3 v:4,a:4” \
“prueba comandos/crf/stream_%v.m3u8”
```

=== VOOP ENCODING ===

Note the only 2 new arguments in the command line, in yellow below:

```
ffmpeg_VOOP -hide_banner -benchmark -nostdin -y -xerror \
-vo_optimizer -vo_vmaf 96 \
-i “promoTivify1.mp4” -max_muxing_queue_size 4096 \
-filter_complex “\
[0:v:0]scale=iw*sar:ih,yadif[v_step_0]; \
[v_step_0]null[v_step_1]; \
[v_step_1]format=yuv420p,split=5[v_step_2_0][v_step_2_1][v_step_2_2][v_step_2_3][v_step_2_4]; \
[v_step_2_0]scale=1920:1080:force_original_aspect_ratio=decrease,pad=1920:1080:-1:-
1:color=black[v_out_0]; \
[v_step_2_1]scale=1280:720:force_original_aspect_ratio=decrease,pad=1280:720:-1:-
1:color=black[v_out_1]; \
[v_step_2_2]scale=854:480:force_original_aspect_ratio=decrease,pad=854:480:-1:-1:color=black[v_out_2];
\
[v_step_2_3]scale=640:360:force_original_aspect_ratio=decrease,pad=640:360:-1:-1:color=black[v_out_3];
\
[v_step_2_4]scale=426:240:force_original_aspect_ratio=decrease,pad=426:240:-1:-1:color=black[v_out_4];
\
```

```
[0:a:0]loudnorm=I=-23:LRA=10:TP=-1[a_step_0]; \
[a_step_0]asplit=5[a_out_0][a_out_1][a_out_2][a_out_3][a_out_4]" \
\
-map "[v_out_0]" -map "[a_out_0]" -c:v:0 libx264 -preset slow \
-x264-params "nal-hrd=none:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50" \
-force_key_frames "expr:gte(t,n_forced*5.0)" -r 25 -profile:v high -bf 2 \
-maxrate:v:0 5161200 -bufsize:v:0 10322400 \
-c:a:0 aac -b:a:0 128k -ac:0 2 -ar 48000 -metadata:s:a:0 language=spa \
\
-map "[v_out_1]" -map "[a_out_1]" -c:v:1 libx264 -preset slow \
-x264-params "nal-hrd=none:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50" \
-force_key_frames "expr:gte(t,n_forced*5.0)" -r 25 -profile:v high -bf 2 \
-maxrate:v:1 2851200 -bufsize:v:1 5702400 \
-c:a:1 aac -b:a:1 128k -ac:1 2 -ar 48000 -metadata:s:a:1 language=spa \
\
-map "[v_out_2]" -map "[a_out_2]" -c:v:2 libx264 -preset slow \
-x264-params "nal-hrd=none:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50" \
-force_key_frames "expr:gte(t,n_forced*5.0)" -r 25 -profile:v high -bf 2 \
-maxrate:v:2 1861200 -bufsize:v:2 3722400 \
-c:a:2 aac -b:a:2 128k -ac:2 2 -ar 48000 -metadata:s:a:2 language=spa \
\
-map "[v_out_3]" -map "[a_out_3]" -c:v:3 libx264 -preset slow \
-x264-params "nal-hrd=none:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50" \
-force_key_frames "expr:gte(t,n_forced*5.0)" -r 25 -profile:v high -bf 2 \
-maxrate:v:3 926200 -bufsize:v:3 1852400 \
-c:a:3 aac -b:a:3 128k -ac:3 2 -ar 48000 -metadata:s:a:3 language=spa \
\
-map "[v_out_4]" -map "[a_out_4]" -c:v:4 libx264 -preset slow \
-x264-params "nal-hrd=none:force-cfr=1:scenecut=-1:min-keyint=50:keyint=50" \
-force_key_frames "expr:gte(t,n_forced*5.0)" -r 25 -profile:v high -bf 2 \
-maxrate:v:4 486200 -bufsize:v:4 972400 \
-c:a:4 aac -b:a:4 128k -ac:4 2 -ar 48000 -metadata:s:a:4 language=spa \
\
-f hls -hls_time 5 -hls_playlist_type vod \
-hls_flags independent_segments+discont_start+program_date_time \
-hls_segment_type mpegts \
-hls_segment_filename "prueba comandos/voop/data_%v_%02d.ts" \
-master_pl_name master.m3u8 \
-var_stream_map "v:0,a:0 v:1,a:1 v:2,a:2 v:3,a:3 v:4,a:4" \
"prueba comandos/voop/stream_%v.m3u8"
```