

Compression Ratio Analysis for Surveillance Video Analytics Datasets Using H.264 and H.265 Standards

Kholidiyah Masykuroh^{1*}

¹Telecommunication Engineering Study Program, Telkom University

¹Jalan D.I Panjaitan No. 128, Purwokerto, 53147, Jawa Tengah, Indonesia

Email : *kholidiyahma@telkomuniversity.ac.id

ARTICLE INFORMATION

Received 20 May 2026

Revised 4 June 2026

Accepted 23 June 2024

Keywords :

Bitstream

Quantization

Video Analytics

Video Compression

Video Encoder

ABSTRACT

This study aims to analyze the effects of Quantization Parameter (QP) values and encoding presets on the compression ratio of surveillance video datasets using the H.264 and H.265 standards for video analytics applications. The increasing use of video services on the internet has made compression efficiency an important aspect, particularly for surveillance applications that require continuous video transmission with efficient bandwidth usage. This study employs FFmpeg on Ubuntu to evaluate two QP values (26 and 36) and two encoding presets (veryslow and ultrafast). The experiments were conducted on 10 videos with varying characteristics, obtained from several sources, including CCTV recordings, website-based video samples, self-recorded videos, and videos from the MATLAB library. The results show that the higher QP value, QP 36, produces a smaller bitstream size and a higher compression ratio than QP 26. In addition, the veryslow preset generally achieves better compression ratios than the ultrafast preset for both compression standards. The Constant Rate Factor (CRF)-based encoding test with a CRF value of 36 also demonstrates competitive compression performance compared with QP-based encoding. Overall, H.265 provides better compression ratios than H.264 in most experimental scenarios. These findings indicate that the combination of H.265, QP 36, and the veryslow preset can be an effective configuration for reducing bandwidth requirements in surveillance video analytics applications, particularly when bitstream size efficiency is the primary priority.

1. Introduction

The increasing number of video service users on the internet has led to a growing need for efficient video compression (Arora et al., 2016; O'Byrne et al., 2022; Zhong & Reibman, 2018). Recent network traffic reports show that video remains a dominant contributor to mobile data traffic. Ericsson reported that, at the end of 2025, video traffic accounted for 76% of all mobile data traffic (Ericsson, 2025). Efficient video compression is therefore crucial for reducing bandwidth usage and enhancing the user experience across various applications, including smart cities, security surveillance, smart homes, retail businesses, and autonomous vehicles (Sieber et al., 2015).

The most commonly used video compression technologies today are H.264 and H.265/HEVC (Reich et al., 2024). These standards have evolved and offer substantial bitrate reductions while maintaining high image quality (Usman et al., 2017). However, with the increasing demand for high-quality video, higher resolutions, and continuous video transmission in surveillance systems, there are still challenges in determining compression configurations that can reduce bitstream size while maintaining sufficient visual quality for video analytics applications (Rippel et al., 2019).

In video compression, quantization plays an important role in controlling the trade-off between bitrate reduction and visual quality. Quantization reduces the precision of transformed video data, thereby decreasing the bitstream size, but excessive quantization may introduce visual distortion and reduce the usefulness of

compressed videos for analytics tasks. This issue is particularly important in surveillance video analytics because compression settings can affect the performance of object detection and monitoring systems (M.-J. Kim & Lee, 2022; O'Byrne et al., 2022). In H.264 and H.265 architectures, the Quantization Parameter (QP) controls the quantization level and directly influences bitrate, output file size, and compression efficiency.

Although many studies have been conducted to improve video compression techniques, most existing work focuses on general compression efficiency, video quality assessment, or codec performance across broad multimedia scenarios. There is still a need for a more specific empirical analysis of how quantization values and encoding presets affect compression ratios in surveillance video analytics datasets (Rerabek et al., 2015; Valin & Terriberry, 2015). This research aims to fill this gap by evaluating various quantization values and compression presets and comparing the results obtained using FFmpeg on the Ubuntu operating system (Otani et al., 2022).

This research aims to evaluate and compare the compression ratios achieved with the H.264 and H.265 standards across different types of surveillance video data, including CCTV recordings and videos from the MATLAB library. The findings of this study are expected to provide new insights for developers and researchers in the field of video analytics and contribute to the development of more efficient compression techniques (Alam et al., 2021; Xu et al., 2018).

The research objectives of this study are threefold: first, to analyze the impact of QP values on the compression ratio of H.264 and H.265; second, to compare the compression performance of H.264 and H.265 on surveillance video datasets; and third, to provide recommendations for selecting effective compression settings for video analytics applications. These objectives are intended to support practical decision-making when choosing codecs, QPs, and preset configurations to reduce data size and bandwidth usage in real-world surveillance scenarios.

The remainder of this paper is organized as follows. The Literature Review section discusses video compression standards, quantization, and related studies in video analytics. The Research Method section describes the dataset, encoding configurations, and evaluation metrics used in the experiments. The Result and Discussion section presents the compression performance of H.264 and H.265 under different QP and preset settings. Finally, the Conclusion section summarizes the main findings and provides recommendations for future research.

2. Literature Review

2.1 Video Compression Standards for Video Analytics

Video compression is a critical technology that enables efficient storage and transmission of video data by reducing file size without compromising quality (Choi & Bajic, 2022; Galteri et al., 2018). In video analytics applications, compression is particularly important because video streams are often transmitted continuously from cameras to processing systems through bandwidth-limited networks. Applications such as smart cities, traffic monitoring, security surveillance, retail monitoring, and autonomous systems require efficient video transmission while maintaining sufficient visual information for analysis.

Among widely adopted video compression standards, H.264/Advanced Video Coding (AVC) and H.265/High Efficiency Video Coding (HEVC) are commonly used for their ability to reduce bitrate while preserving acceptable video quality. H.264 has been extensively implemented in various video applications because it provides good compression performance at relatively low bitrates (I.-K. Kim et al., 2014). H.265/HEVC was developed as a successor to H.264 and offers higher compression efficiency, including support for ultra-high-definition video and improved coding structures (Uhrina et al., 2014).

The evolution of video compression standards began with earlier techniques such as MPEG-1 and MPEG-2, which laid the foundation for modern video coding systems. H.264/AVC was developed as a joint standardization project between the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving

Picture Experts Group (MPEG), while H.265/HEVC was developed by the Joint Collaborative Team on Video Coding (JCT-VC) (Fraunhofer Heinrich Hertz Institute, 2021; ISO/IEC, 2020; ITU-T, 2024; Sullivan et al., 2012; Wiegand et al., 2003). Compared with H.264, HEVC introduces more flexible block partitioning, improved prediction, transform coding, and entropy coding mechanisms, which contribute to its higher compression efficiency.

2.2 Quantization Parameter and Compression Trade-Off

Quantization is a fundamental process in video compression that reduces the precision of transformed video data to decrease the number of bits required for representation (Mao et al., 2024). This process directly affects the trade-off between bitstream size and video quality. Stronger quantization can reduce bitrate and increase the compression ratio, but it may also introduce quantization noise and visual distortion.

In H.264 and H.265, the QP controls the quantization step size. A higher QP value generally increases the quantization step size, resulting in a smaller bitstream and a higher compression ratio. However, this may reduce visual quality because more information is discarded during encoding. Conversely, a lower QP value usually produces better visual quality but results in a larger bitstream. Therefore, QP selection is important for balancing compression efficiency and video quality.

Quantization can be implemented in different forms, including scalar and vector quantization. In most image and video coding standards, scalar quantization is commonly used together with transform coding (Metkar & Talbar, 2013). The quantization process is irreversible because several input values may be mapped into a smaller set of output values. As a result, compression efficiency improves, but some information from the original video may be lost. This trade-off is important in video analytics because excessive compression can reduce the visual detail required for detection, tracking, or monitoring tasks.

2.3 Learning-Based and Perceptual Quantization Approaches

Recent advancements in video compression have explored learning-based techniques and perceptual video coding to improve compression efficiency while maintaining perceptual quality. Learning-based methods, including convolutional neural networks (CNNs), have been investigated to optimize quantization and improve bitrate reduction with limited quality degradation. Ki et al. studied learning-based just-noticeable-quantization-distortion models for perceptual video coding and showed that quantization-related optimization can reduce bitrate while maintaining subjective video quality (Ki et al., 2018).

Li et al. proposed a Trellis-Coded Quantizer (TCQ) for deep learning-based image compression and compared its performance with scalar quantization (Li et al., 2020). Although their work focuses on learned image compression rather than standard H.264/H.265 video encoding, it demonstrates that quantization design plays an important role in improving compression efficiency. These studies indicate that quantization remains a relevant research topic in both traditional video coding standards and emerging learning-based compression approaches.

However, learning-based compression approaches often incur additional computational overhead and require model-specific implementations. In practical surveillance video analytics systems, standard codecs such as H.264 and H.265 remain widely used because of their compatibility with existing cameras, video recorders, and network transmission systems. Therefore, empirical evaluation of QP and encoding preset configurations in these standards remains relevant for practical deployment.

2.4 Video Compression in Surveillance Video Analytics

Video analytics refers to the process of extracting useful information from video data, such as detecting objects, counting people, monitoring traffic, or identifying activities. In modern systems, video analytics is

commonly supported by computer vision and artificial intelligence techniques. These applications are widely used in smart cities, traffic control, crowd monitoring, security surveillance, industrial environments, and public safety systems.

Several video analytics systems have been developed to improve the efficiency of video transmission and analysis. AWStream, CloudSeg, DDS, and AccMPEG are examples of systems that consider the relationship between video transmission, network resources, and analytics performance (Du et al., 2020, 2022; Wang et al., 2019; Zhang et al., 2018). These studies show that video analytics performance is not only influenced by the accuracy of the analytics model but also by the quality and bitrate of the input video stream.

In surveillance video analytics, compression must be carefully configured because the goal is not only to reduce file size but also to preserve useful visual information. Strong compression may reduce bandwidth requirements, but excessive compression can remove important spatial or temporal details that are needed for analysis. Therefore, codec selection, QP values, and encoding presets need to be evaluated in relation to compression efficiency and practical surveillance requirements.

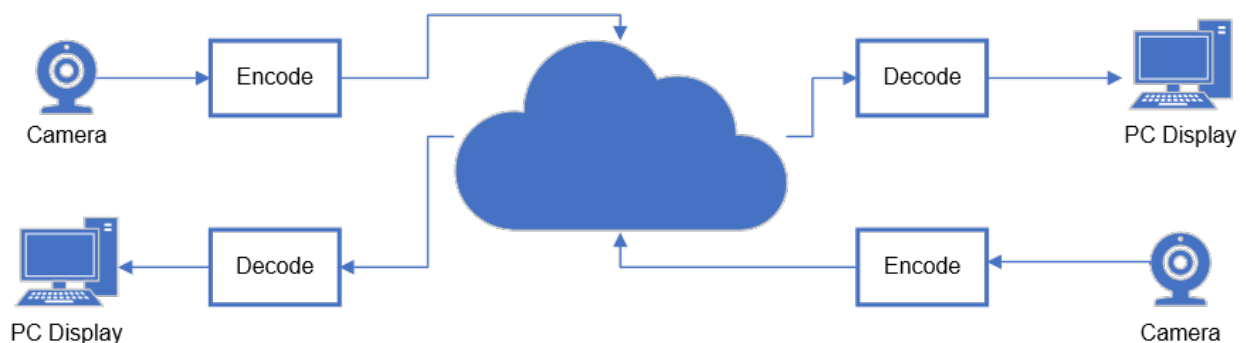


Figure 1. Two-Way Video Coding Scenario (Richardson, 2011).

Fig. 1 illustrates a general two-way video coding and transmission scenario. Video data captured by a camera is encoded before being transmitted through a network. At the receiver side, the encoded bitstream is decoded and displayed for further use. This general workflow highlights the importance of video encoding in reducing bitstream size before transmission, particularly in bandwidth-limited video analytics applications.

2.5 H.264 and H.265 Coding Structures

H.264/AVC follows a hybrid video coding framework that combines prediction, transform coding, quantization, and entropy coding. The standard exploits temporal, spatial, visual, and statistical redundancies to reduce video size. Temporal redundancy is reduced by using similarities between adjacent frames, while spatial redundancy is reduced by exploiting correlations among neighboring pixels. Visual redundancy is related to information that is less noticeable to the human visual system, and statistical redundancy is reduced through entropy coding.

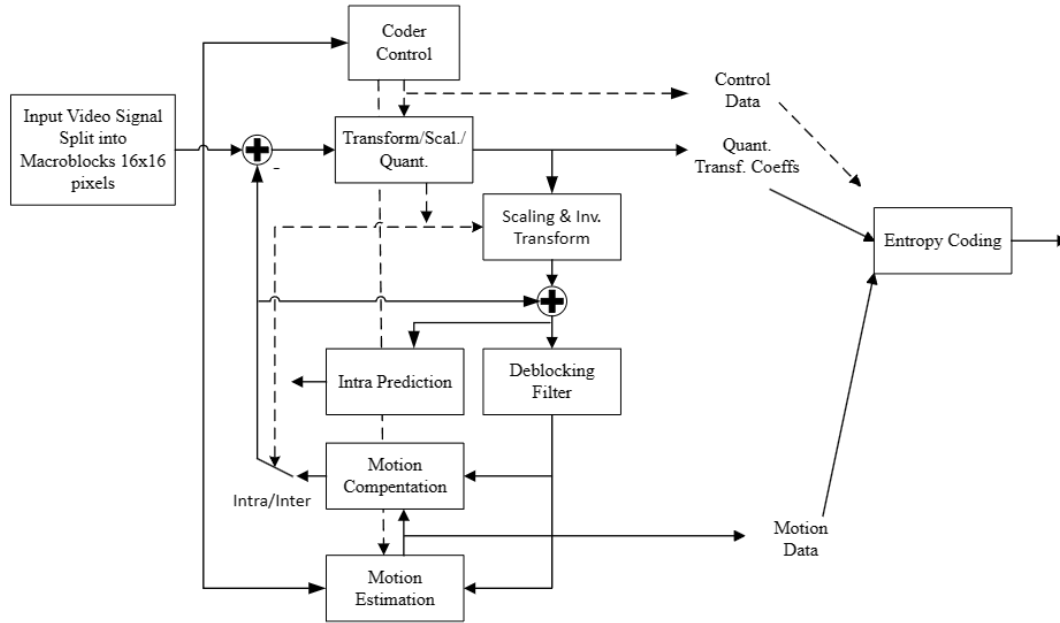


Figure 2. H.264 framework (Masykuroh et al., 2025; Wiegand et al., 2003).

The ISO/IEC JTC1 Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) collaborated to produce the H.264/MPEG-4 Advanced Video Coding (AVC) standard for video coding. This collaboration led to the formation of the Joint Video Team (JVT). As a result of this joint development, the technical content of the ISO/IEC MPEG-4 AVC standard, formally known as ISO/IEC 14496-10 MPEG-4 Part 10, and the ITU-T H.264 standard is identical. Although H.264 follows a hybrid video coding framework, as shown in Fig. 2, several coding tools have been improved to increase compression efficiency (Wu, 2014).

All image and video compression schemes are designed to exploit several types of redundancy. The goal is to minimize distortion at a given compression rate or to minimize the compression rate for a given distortion. These redundancy types explain why video coding standards can reduce bitstream size and why different coding structures may produce different compression ratios. The four main types of redundancy are:

1. Temporal redundancy — Adjacent frames often look similar. Differences between frames are frequently caused by camera movement or changes in scene elements. By exploiting similarities between consecutive frames, video encoders can reduce repeated information.
2. Spatial redundancy — Neighboring pixels in an image often have correlated values because objects and backgrounds usually contain consistent texture patterns. This correlation allows image content to be represented more efficiently in the frequency domain.
3. Visual redundancy — The human visual system (HVS) is less sensitive to certain losses in visual data. This allows video encoders to remove or reduce information that is less perceptually significant while maintaining acceptable visual quality.
4. Statistical redundancy — Video data often contains symbols with non-uniform probability distributions. Entropy coding exploits this statistical property to represent frequent symbols with fewer bits.

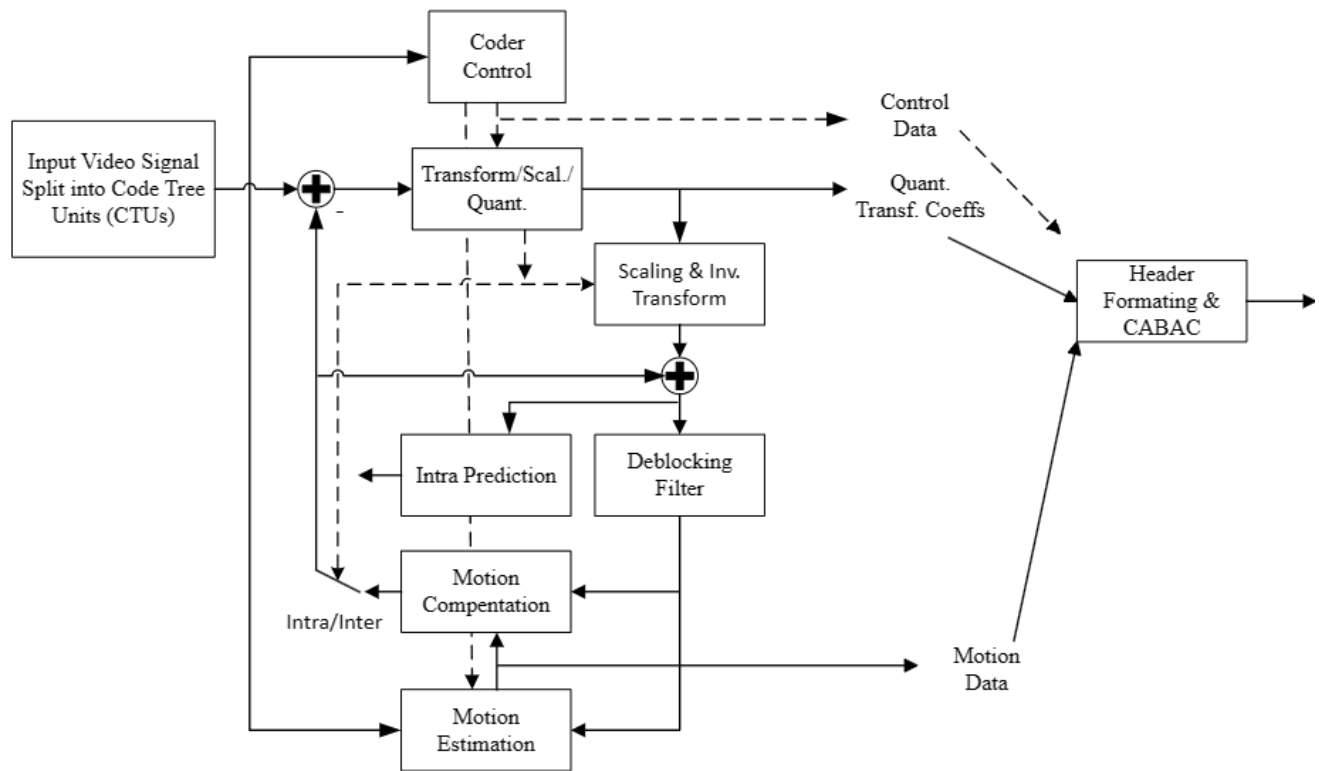


Figure 3. H.265 standard framework (Masykuroh et al., 2025; Sullivan et al., 2012).

As ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265, High Efficiency Video Coding (HEVC) is a video compression standard jointly developed by ISO/IEC JTC1 MPEG and ITU-T VCEG. It was developed as the successor to H.264/MPEG-4 AVC. To create the HEVC standard, MPEG and VCEG established the Joint Collaborative Team on Video Coding (JCT-VC). Compared with H.264/MPEG-4 AVC, HEVC can achieve higher compression efficiency and support ultra-high-definition (UHD) video compression, including resolutions up to 8192×4320 (Wu, 2014).

The HEVC standard replaces the macroblock structure used in earlier standards with a more flexible block-partitioning method. HEVC divides a frame into Coding Tree Units (CTUs), which can be further subdivided into Coding Tree Blocks (CTBs) for each luma and chroma component. CTBs may have block sizes of 64×64 , 32×32 , or 16×16 , where larger block sizes often improve coding efficiency. CTBs are then divided into Coding Units (CUs), which can be recursively split using a quadtree structure. Prediction Units (PUs) are used for intra-frame and inter-frame prediction, with sizes ranging from 64×64 to 4×4 (Wu, 2014).

The structural differences between H.264 and H.265 are important in this study because they may affect the resulting bitstream size and compression ratio under the same QP and preset configurations. Therefore, both standards are evaluated using identical experimental settings to identify practical compression configurations for surveillance video analytics.

2.6 Research Gap

Although many studies have investigated video compression standards, quantization, perceptual quality, and learning-based compression, most existing work focuses on either general compression efficiency or visual quality assessment. Fewer studies provide a focused empirical comparison of H.264 and H.265 under different

QP values, CRF-based encoding, and encoding presets using surveillance-oriented video analytics datasets. This gap is important because surveillance video analytics requires compression configurations that can reduce bitstream size while preserving sufficient visual information for monitoring and analysis.

Therefore, this study evaluates the compression ratios of H.264 and H.265 across different QP values and encoding presets on surveillance video datasets. By comparing the performance of both standards under controlled encoding configurations, this study provides practical insight into selecting efficient compression settings for surveillance video analytics applications.

3. Research Method

This section describes the experimental methodology used to evaluate the effects of QP values and encoding presets on the compression performance of surveillance video datasets encoded using H.264 and H.265 standards. The methodology was designed to compare how different encoding configurations influence output file size, bitrate, and compression ratio. The broader objective of this experimental design is to support bandwidth-efficient video analytics applications by identifying practical compression configurations for transmitting surveillance video.

FFmpeg was used as the video encoding tool on Ubuntu to provide a consistent, reproducible experimental environment. The experiments were conducted using the same encoding configurations for all selected videos. Each video was encoded using both H.264 and H.265 standards, with two QP values and two encoding presets. This procedure allows a fair comparison between the two video coding standards under identical experimental conditions.

3.1 Experimental Design

The experimental design was developed to compare the compression performance of H.264 and H.265 under different encoding configurations. Two QP values, namely 26 and 36, were selected to represent different compression levels. QP 26 was used to represent a lower quantization level, which generally preserves more visual information but produces a larger output file. QP 36 was used to represent a higher quantization level, which generally produces a smaller bitstream but may introduce more visual distortion.

In addition to QP values, two encoding presets, namely *veryslow* and *ultrafast*, were used to evaluate the effect of encoding complexity on compression performance. The *veryslow* preset allows the encoder to perform more extensive optimization during encoding, whereas the *ultrafast* preset prioritizes faster encoding with less compression optimization. CRF-based encoding with a CRF value of 36 was also evaluated as an additional comparison to the QP-based encoding configurations.

Each selected video was encoded using both H.264 and H.265 standards. For each standard, the same QP values and encoding presets were applied to ensure consistency across all experimental scenarios. This design enables a controlled comparison of codec performance, QP effects, and preset effects on surveillance video compression.

3.2 Dataset Description

The dataset used in this study consists of ten surveillance-related videos obtained from several sources, including CCTV recordings, videos from the MATLAB library, publicly available website-based video samples, and self-recorded videos. These videos were selected to represent different surveillance conditions, including variations in scene content, duration, motion characteristics, and recording sources. The use of heterogeneous video sources is intended to provide a broader evaluation of compression performance across different surveillance scenarios.

The dataset includes videos with varying durations and recording characteristics to reflect practical conditions for surveillance video analytics. Detailed information on the video dataset, including video name, source, duration, and recording category, is presented in the experimental result tables.

3.3 Encoding Configuration

FFmpeg was used as the video encoding tool on Ubuntu to ensure a consistent and reproducible experimental environment (Van Sickel, 2024). The encoding parameters consisted of two video coding standards, two QP values, two encoding presets, and one CRF-based configuration. Instead of presenting command-line instructions, the encoding settings used in this study are summarized in Table 1.

Table 1. Experimental Encoding Configuration (FFmpeg, 2025).

Parameter	Configuration
Encoding tool	FFmpeg
Operating system	Ubuntu
Video coding standards	H.264 and H.265
Encoding mode	QP-based encoding and CRF-based encoding
QP values	26 and 36
CRF value	36
Encoding presets	very slow and ultrafast
Number of video samples	10 videos
Video sources	CCTV recordings, MATLAB library videos, website-based video samples, and self-recorded videos
Main evaluation metrics	Output file size, bitrate, and compression ratio
Additional evaluation	General visual inspection of compressed videos
Purpose of evaluation	To compare compression performance under different codecs, QP, CRF, and preset configurations

These configurations were applied consistently across all selected videos to ensure a fair comparison between H.264 and H.265 under identical experimental conditions. Using the same dataset and parameter combinations allows the effects of the codec standard, QP value, and encoding preset to be observed more clearly.

3.4 Experimental Workflow

Fig. 4 illustrates the experimental workflow used to evaluate the compression performance of H.264 and H.265 on surveillance video datasets. The workflow begins with selecting surveillance-related videos from various sources. The selected videos are then encoded with FFmpeg using two video coding standards: H.264 and H.265. For each standard, the encoding process is configured using two presets, namely veryslow and ultrafast, and two QP values, namely 26 and 36.

The encoding procedure is repeated for each combination of video, codec standard, QP value, and encoding preset. This repeated process ensures that all experimental scenarios are evaluated consistently. The compressed video files generated by the encoding process are then analyzed for output file size, bitrate, and compression ratio. In addition, a visual inspection is conducted to assess the overall quality of the compressed videos.

Based on these evaluations, the study identifies the trade-off between bitstream reduction and visual quality preservation. The resulting analysis is used to derive recommended encoding configurations for bandwidth-efficient surveillance video analytics applications.

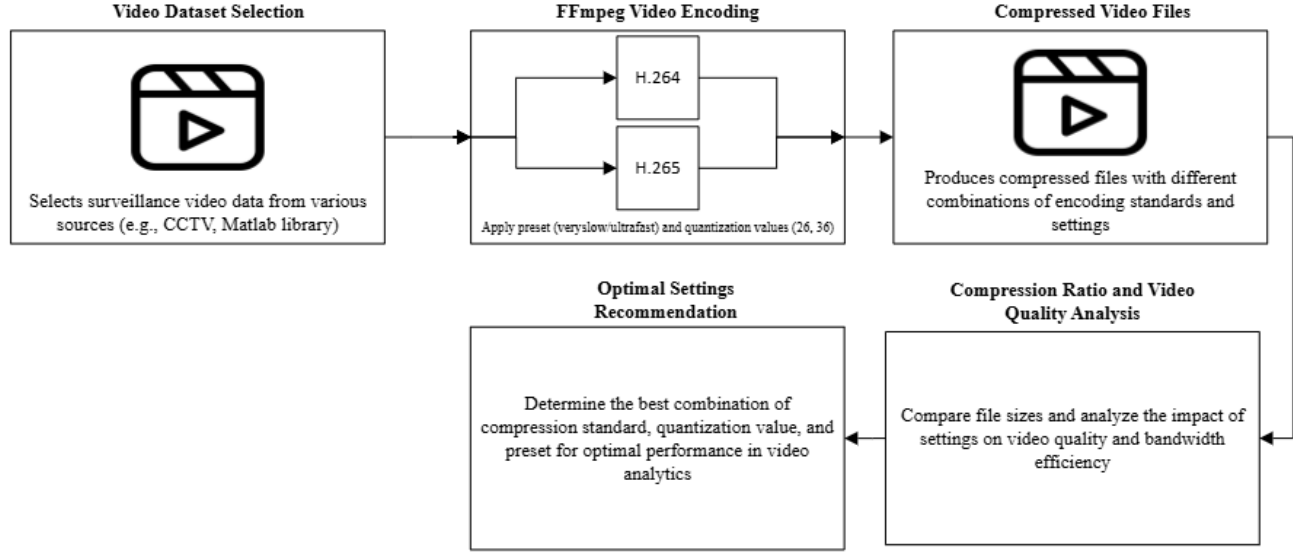


Figure 4. Block Diagram: Video Compression and Analysis Workflow.

3.5 Evaluation Metrics

The main evaluation metric used in this study is the compression ratio. In this study, the compression ratio is expressed as the percentage reduction in file size after encoding. It is calculated using Eq. (1):

$$\text{Compression Ratio (\%)} = \frac{\text{Original File Size} - \text{Compressed File Size}}{\text{Original File Size}} \times 100\% \quad 1)$$

This metric represents the percentage reduction in file size achieved after encoding, relative to the original video size. A higher compression ratio indicates that the encoded video requires a smaller bitstream, reflecting more efficient compression.

In addition to the compression ratio, the output file size and bitrate were analyzed to evaluate the effect of each encoding configuration. The analysis also considers the relationships among QP values, bitrate, and the overall visual characteristics of the compressed videos. Higher QP values are expected to reduce bitstream size by increasing the quantization step size, whereas lower QP values tend to preserve more visual information but produce larger output files. This relationship provides the basis for evaluating the trade-off between compression efficiency and the preservation of visual quality.

The selected encoding configurations, including the codec standard, QP value, encoding preset, and CRF value, are evaluated for their effects on output file size, bitrate, and compression ratio. This evaluation setup enables a controlled comparison of H.264 and H.265 compression performance on surveillance video analytics datasets.

4. Results and Discussion

This section presents and discusses the compression performance of H.264 and H.265 under different encoding configurations. The evaluation is based on output file size, bitrate, and compression ratio obtained from ten surveillance-related videos. The experiments compare QP-based encoding at QP 26 and QP 36 using two encoding presets: veryslow and ultrafast. In addition, CRF-based encoding with a CRF value of 36 is evaluated for comparison.

The discussion focuses on four main aspects: the effects of QP values, encoding presets, and CRF-based encoding; the comparison between H.264 and H.265; and the practical implications of the results for surveillance video analytics applications.

4.1 H.264 Compression Results

Tables 2–6 present the compression results obtained using the H.264/AVC standard under different encoding configurations. The results show that the QP value influences the output file size and compression ratio. In general, QP 36 produces a smaller bitstream and a higher compression ratio than QP 26. This is expected because a higher QP value increases the quantization step size, resulting in stronger compression and lower output bitrate.

Table 2. H.264 Compression Results Using the veryslow Preset and QP 36.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	36	H.264	339	2.712	59,45%
2	parking	Website	02:29	77.968	36	H.264	587	4.696	93,98%
3	pedestrian	Website	00:14	57.544	36	H.264	890	7.120	87,63%
4	vipmen	Matlab Library	00:09	128.464	36	H.264	33	264	99,79%
5	virat	Website	00:24	41.008	36	H.264	417	3.336	91,87%
6	walking	Website	00:09	11.192	36	H.264	370	2.960	73,55%
7	cctv_1	CCTV	05:28	434.920	36	H.264	4.448	35.584	91,82%
8	cctv_2	CCTV	17:25	1.395.504	36	H.264	13.004	104.032	92,55%
9	record_1	self-recorded	05:02	5.083.864	36	H.264	52.720	421.760	91,70%
10	Record 2	self-recorded	02:13	2.250.640	36	H.264	22.269	178.152	92,08%

Table 3. H.264 Compression Results Using the veryslow Preset and QP 26.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	26	H.264	803	6.424	3,95%
2	parking	Website	02:29	77.968	26	H.264	1.970	15.760	79,79%
3	pedestrian	Website	00:14	57.544	26	H.264	2.669	21.352	62,89%
4	vipmen	Matlab Library	00:09	128.464	26	H.264	116	928	99,28%
5	virat	Website	00:24	41.008	26	H.264	1.804	14.432	64,81%
6	walking	Website	00:09	11.192	26	H.264	1.124	8.992	19,66%
7	cctv_1	CCTV	05:28	434.920	26	H.264	14.880	119.040	72,63%
8	cctv_2	CCTV	17:25	1.395.504	26	H.264	46.145	369.160	73,55%
9	record_1	self-recorded	05:02	5.083.864	26	H.264	341.582	2.732.656	46,25%
10	Record 2	self-recorded	02:13	2.250.640	26	H.264	134.893	1.079.144	52,05%

Table 4. H.264 Compression Results Using the ultrafast Preset and QP 36.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	36	H.264	638	5.104	23,68%
2	parking	Website	02:29	77.968	36	H.264	667	5.336	93,16%
3	pedestrian	Website	00:14	57.544	36	H.264	1.008	8.064	85,99%
4	vipmen	Matlab Library	00:09	128.464	36	H.264	50	400	99,69%
5	virat	Website	00:24	41.008	36	H.264	491	3.928	90,42%
6	walking	Website	00:09	11.192	36	H.264	575	4.600	58,90%
7	cctv_1	CCTV	05:28	434.920	36	H.264	5.328	42.624	90,20%
8	cctv_2	CCTV	17:25	1.395.504	36	H.264	15.026	120.208	91,39%
9	record_1	self-recorded	05:02	5.083.864	36	H.264	207.835	1.662.680	67,29%
10	record_2	self-recorded	02:13	2.250.640	36	H.264	71.937	575.496	74,43%

Table 5. H.264 Compression Results Using the ultrafast Preset and QP 26.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	26	H.264	638	5.104	23,68%
2	parking	Website	02:29	77.968	26	H.264	2.409	19.272	75,28%
3	pedestrian	Website	00:14	57.544	26	H.264	3.824	30.592	46,84%
4	vipmen	Matlab Library	00:09	128.464	26	H.264	184	1.472	98,85%
5	virat	Website	00:24	41.008	26	H.264	2.237	17.896	56,36%
6	walking	Website	00:09	11.192	26	H.264	3.088	24.704	-120,73%
7	cctv_1	CCTV	05:28	434.920	26	H.264	18.415	147.320	66,13%
8	cctv_2	CCTV	17:25	1.395.504	26	H.264	55.152	441.216	68,38%
9	record_1	self-recorded	05:02	5.083.864	26	H.264	1.345.043	10.760.344	-111,66%
10	record_2	self-recorded	02:13	2.250.640	26	H.264	482.620	3.860.960	-71,55%

Table 6. H.264 Compression Results Using CRF 36.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	CRF	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	36	H.264	207	1.656	75,24%
2	parking	Website	02:29	77.968	36	H.264	814	6.512	91,65%
3	pedestrian	Website	00:14	57.544	36	H.264	749	5.992	89,59%
4	vipmen	Matlab Library	00:09	128.464	36	H.264	22	176	99,86%
5	virat	Website	00:24	41.008	36	H.264	375	3.000	92,68%
6	walking	Website	00:09	11.192	36	H.264	253	2.024	81,92%
7	cctv_1	CCTV	05:28	434.920	36	H.264	4.343	34.744	92,01%
8	cctv_2	CCTV	17:25	1.395.504	36	H.264	12.869	102.952	92,62%
9	record_1	self-recorded	05:02	5.083.864	36	H.264	36.137	289.096	94,31%
10	record_2	self-recorded	02:13	2.250.640	36	H.264	15.783	126.264	94,39%

For H.264 using the veryslow preset, QP 36 achieves an average compression ratio of approximately 87.44%, while QP 26 achieves approximately 57.49%. This indicates that increasing the QP value from 26 to 36 substantially reduces the output file size. The same tendency is also observed in the ultrafast preset, where QP 36 produces higher and more stable compression ratios than QP 26.

The results also show that the veryslow preset generally provides better compression efficiency than the ultrafast preset. This occurs because the veryslow preset allows the encoder to perform more extensive optimization during encoding. Although this preset requires longer encoding time, it can produce smaller output files and higher compression ratios. In contrast, the ultrafast preset prioritizes encoding speed, which may result in larger output files and lower compression efficiency.

Several negative compression ratios are observed in the H.264 ultrafast QP 26 configuration, particularly for some video samples such as walking, record_1, and record_2. A negative compression ratio indicates that the encoded file size exceeds the original file size, suggesting that the selected encoding configuration is inefficient for that particular video sample. This condition may occur when the selected encoding configuration is inefficient for a particular video source, especially when the original video has already been compressed or when the preset does not provide sufficient compression.

The CRF-based encoding with CRF 36 also shows strong compression performance for H.264, with an average compression ratio of approximately 90.43%. This result indicates that CRF-based encoding can provide competitive compression efficiency compared with QP-based encoding, particularly when the encoder is allowed to adapt bitrate allocation according to video content complexity.

4.2 H.265 Compression Results

Tables 7–11 present the compression results obtained using the H.265/HEVC standard under different encoding configurations. Similar to the H.264 results, the H.265 results show that the QP value clearly influences the output file size and compression ratio. In general, QP 36 produces smaller output files and higher compression ratios than QP 26. However, the magnitude of improvement varies across video samples, indicating that compression performance is also influenced by video content characteristics, duration, motion complexity, and the original file condition.

Table 7. H.265 Compression Results Using the veryslow Preset and QP 36.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	36	H.265	326	2.608	61,00%
2	parking	Website	02:29	77.968	36	H.265	465	3.720	95,23%
3	pedestrian	Website	00:14	57.544	36	H.265	652	5.216	90,94%
4	vipmen	Matlab Library	00:09	128.464	36	H.265	43	344	99,73%
5	virat	Website	00:24	41.008	36	H.265	349	2.792	93,19%
6	walking	Website	00:09	11.192	36	H.265	308	2.464	77,98%
7	cctv_1	CCTV	05:28	434.920	36	H.265	3.501	28.008	93,56%
8	cctv_2	CCTV	17:25	1.395.504	36	H.265	10.180	81.440	94,16%
9	record_1	self-recorded	05:02	5.083.864	36	H.265	36.396	291.168	94,27%
10	record_2	self-recorded	02:13	2.250.640	36	H.265	15.650	125.200	94,44%

Table 8. H.265 Compression Results Using the veryslow Preset and QP 26.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	26	H.265	878	7.024	-5,02%
2	parking	Website	02:29	77.968	26	H.265	1.672	13.376	82,84%
3	pedestrian	Website	00:14	57.544	26	H.265	2.439	19.512	66,09%
4	vipmen	Matlab Library	00:09	128.464	26	H.265	221	1.768	98,62%
5	virat	Website	00:24	41.008	26	H.265	1.793	14.344	65,02%
6	walking	Website	00:09	11.192	26	H.265	1.113	8.904	20,44%
7	cctv_1	CCTV	05:28	434.920	26	H.265	13.122	104.976	75,86%
8	cctv_2	CCTV	17:25	1.395.504	26	H.265	40.634	325.072	76,71%
9	record_1	self-recorded	05:02	5.083.864	26	H.265	371.265	2.970.120	41,58%
10	record_2	self-recorded	02:13	2.250.640	26	H.265	146.378	1.171.024	47,97%

Table 9. H.265 Compression Results Using the ultrafast Preset and QP 36.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	36	H.265	342	2.736	59,09%
2	parking	Website	02:29	77.968	36	H.265	505	4.040	94,82%
3	pedestrian	Website	00:14	57.544	36	H.265	665	5.320	90,75%
4	vipmen	Matlab Library	00:09	128.464	36	H.265	49	392	99,69%
5	virat	Website	00:24	41.008	36	H.265	339	2.712	93,39%
6	walking	Website	00:09	11.192	36	H.265	297	2.376	78,77%
7	cctv_1	CCTV	05:28	434.920	36	H.265	3.984	104.976	75,86%
8	cctv_2	CCTV	17:25	1.395.504	36	H.265	11.368	325.072	76,71%
9	record_1	self-recorded	05:02	5.083.864	36	H.265	45.756	2.970.120	41,58%
10	record_2	self-recorded	02:13	2.250.640	36	H.265	17.859	1.171.024	47,97%

Table 10. H.265 Compression Results Using the ultrafast Preset and QP 26.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	Quantization	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	26	H.265	1.492	11.936	-78,47%
2	parking	Website	02:29	77.968	26	H.265	1.956	15.648	79,93%
3	pedestrian	Website	00:14	57.544	26	H.265	2.485	19.880	65,45%
4	vipmen	Matlab Library	00:09	128.464	26	H.265	257	2.056	98,40%
5	virat	Website	00:24	41.008	26	H.265	1.740	13.920	66,06%
6	walking	Website	00:09	11.192	26	H.265	1.382	11.056	1,22%
7	cctv_1	CCTV	05:28	434.920	26	H.265	15.191	121.528	72,06%
8	cctv_2	CCTV	17:25	1.395.504	26	H.265	45.087	360.696	74,15%
9	record_1	self-recorded	05:02	5.083.864	26	H.265	371.268	2.970.144	41,58%
10	record_2	self-recorded	02:13	2.250.640	26	H.265	161.218	1.289.744	42,69%

Table 11. H.265 Compression Results Using CRF 36.

No	Video	Video Source	Duration (s)	Original Filesize (Kbps)	CRF	Video Encoder	Output Filesize (KB)	Output Filesize (Kbps)	Compression Ratio
1	crowd	Website	00:13	6.688	36	H.265	253	2.024	69,74%
2	parking	Website	02:29	77.968	36	H.265	680	5.440	93,02%
3	pedestrian	Website	00:14	57.544	36	H.265	587	4.696	91,84%
4	vipmen	Matlab Library	00:09	128.464	36	H.265	27	216	99,83%
5	virat	Website	00:24	41.008	36	H.265	392	3.136	92,35%
6	walking	Website	00:09	11.192	36	H.265	257	2.056	81,63%
7	cctv_1	CCTV	05:28	434.920	36	H.265	3.730	29.840	93,14%
8	cctv_2	CCTV	17:25	1.395.504	36	H.265	10.866	86.928	93,77%
9	record_1	self-recorded	05:02	5.083.864	36	H.265	30.760	246.080	95,16%
10	record_2	self-recorded	02:13	2.250.640	36	H.265	15.783	126.264	94,39%

For H.265 using the veryslow preset, QP 36 achieves an average compression ratio of approximately 89.45%, while QP 26 achieves approximately 57.01%. This result shows that QP 36 provides more aggressive compression than QP 26. A higher compression ratio indicates that H.265 can reduce the bitstream size at higher QP values.

The effect of encoding preset is also observed in the H.265 results. The veryslow preset generally provides higher compression efficiency than the ultrafast preset because it allows more encoding optimization. This is particularly important in bandwidth-limited surveillance applications, where reducing bitstream size may be more important than minimizing encoding time. However, in scenarios requiring real-time encoding, the ultrafast preset may still be useful despite its lower compression efficiency.

The H.265 CRF 36 configuration achieves an average compression ratio of 90.49%, making it one of the strongest configurations for reducing file size. This suggests that CRF-based encoding can effectively reduce file size while allowing the encoder to adjust compression based on the characteristics of the video content.

4.3 Comparison Between H.264 and H.265

The comparison between H.264 and H.265 shows that H.265 generally provides better compression efficiency than H.264 under similar encoding configurations. This result is consistent with the technical characteristics of H.265/HEVC, which uses more flexible block partitioning, improved prediction, and more efficient coding tools compared with H.264/AVC.

Under the veryslow preset with QP 36, H.265 achieves an average compression ratio of approximately 89.45%, while H.264 achieves approximately 87.44%. This indicates that H.265 provides better compression efficiency with the same QP and preset. The difference is also evident in several longer surveillance videos, including CCTV and self-recorded footage, where H.265 tends to produce smaller bitstreams.

However, the results also show that the superiority of H.265 is not always uniform across all video samples and configurations. Some videos show similar compression ratios between H.264 and H.265, while other videos show larger differences. This indicates that compression performance is affected not only by the codec standard but also by video content characteristics, duration, motion complexity, the original compression format, the QP value, and the preset selection.

The comparison also shows that CRF-based encoding provides strong compression performance for both standards. H.264 CRF 36 and H.265 CRF 36 achieve average compression ratios of approximately 90.43% and

90.49%, respectively. This suggests that CRF-based encoding may be a practical alternative when adaptive bitrate allocation is preferred over fixed QP-based encoding.

4.4 Effect of QP, Preset, and CRF on Compression Performance

The results confirm that the QP value is one of the most important factors affecting compression performance. A higher QP value, such as QP 36, produces a smaller bitstream and a higher compression ratio because more information is discarded during quantization. Conversely, QP 26 tends to preserve more visual information but produces larger output files. Therefore, the selection of the QP value should consider the trade-off between file-size reduction and visual quality preservation.

The encoding preset also affects compression performance. The veryslow preset generally produces better compression ratios than the ultrafast preset because it performs more extensive encoding optimization. However, this improvement comes at the cost of longer encoding time. For offline video storage or non-real-time surveillance data processing, the veryslow preset may be more suitable. For real-time or low-latency applications, the ultrafast preset may still be considered when encoding speed is more important than maximum compression efficiency.

CRF-based encoding shows competitive performance compared with QP-based encoding. The CRF 36 setting achieves high compression ratios for both H.264 and H.265, indicating that CRF can effectively adapt compression to content complexity. This makes CRF-based encoding useful when a balance between compression efficiency and general visual quality is needed.

4.5 Practical Implications for Surveillance Video Analytics

The findings have practical implications for surveillance video analytics applications. In systems where bandwidth efficiency and storage reduction are the main priorities, H.265 with QP 36 and the veryslow preset can be considered an effective configuration, particularly for offline or non-real-time processing. This configuration delivers strong compression performance and reduces the bitstream size.

For applications that require faster encoding, the ultrafast preset can be used, though it may yield lower compression efficiency. Meanwhile, QP 26 may be more appropriate when preserving more visual information is important, but it requires larger storage and bandwidth. CRF 36 can also be considered as an alternative configuration because it provides strong compression performance across different video samples.

Overall, the results indicate that the compression configuration should be selected based on the surveillance system's operational requirements. If the main priority is reducing bandwidth and storage usage, higher QP values and H.265-based encoding are preferable. If visual quality preservation or real-time encoding is more important, lower QP values or faster presets may be selected. Therefore, the choice of codec, QP value, preset, and CRF setting should take into account the trade-offs among compression efficiency, visual quality, and processing requirements.

5. Conclusion

This study evaluated the effect of QP values and encoding presets on the compression performance of H.264 and H.265 standards using surveillance video analytics datasets. The experiments were conducted on ten surveillance-related videos with different characteristics by comparing QP-based encoding at QP 26 and QP 36, two encoding presets, namely veryslow and ultrafast, and CRF-based encoding with a CRF value of 36.

The results show that the QP value influences the output file size, bitrate, and compression ratio. In general, QP 36 produces smaller bitstream sizes and higher compression ratios than QP 26. This indicates that a higher QP value can improve compression efficiency, although it may reduce the amount of visual information preserved in the compressed video. Conversely, QP 26 tends to preserve more visual detail but produces larger output files.

The comparison between H.264 and H.265 shows that H.265 generally achieves better compression efficiency than H.264 under similar encoding configurations. This result is particularly relevant for surveillance video analytics applications, where continuous video transmission can require substantial bandwidth and storage resources. The results also indicate that the veryslow preset generally provides higher compression efficiency than the ultrafast preset because it allows more extensive encoding optimization. However, the ultrafast preset may still be useful in scenarios where encoding speed is more important than maximum compression efficiency.

Overall, the combination of H.265, QP 36, and the veryslow preset can be considered an effective configuration for reducing bitstream size in bandwidth-limited surveillance video analytics applications. CRF-based encoding with CRF 36 also shows competitive compression performance and can be considered as an alternative configuration when adaptive bitrate allocation is required. Nevertheless, the choice of encoding configuration should take into account the specific characteristics of the video content, available bandwidth, storage constraints, and the required level of visual quality.

Future work should extend this evaluation by incorporating objective video quality metrics, such as PSNR, SSIM, and VMAF, as well as video analytics performance metrics, such as object detection accuracy, precision, recall, and mAP. Further studies may also include more recent video coding standards, such as VVC/H.266, and evaluate the impact of compression settings on real-time video analytics systems.

References

- [1] Alam, A., Khalid, S., Khan, M. N., Afridi, T. H., Ullah, I., & Lee, Y.-K. (2021). Video Big Data Analytics in the Cloud: Research Issues and Challenges. In M. Ben Ahmed, İ. Rakıp Karaş, D. Santos, O. Sergeyeva, & A. A. Boudhir (Eds.), *Innovations in Smart Cities Applications Volume 4* (Vol. 183, pp. 1433–1448). Springer International Publishing. https://doi.org/10.1007/978-3-030-66840-2_108
- [2] Arora, S., Bhatia, K., & Amit, V. (2016). Storage optimization of video surveillance from CCTV camera. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 710–713. <https://doi.org/10.1109/NGCT.2016.7877503>
- [3] Choi, H., & Bajic, I. V. (2022). Scalable Video Coding for Humans and Machines. *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, 1–6. <https://doi.org/10.1109/MMSP55362.2022.9949576>
- [4] Du, K., Pervaiz, A., Yuan, X., Chowdhery, A., Zhang, Q., Hoffmann, H., & Jiang, J. (2020). Server-Driven Video Streaming for Deep Learning Inference. *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, 557–570. <https://doi.org/10.1145/3387514.3405887>
- [5] Du, K., Zhang, Q., Arapin, A., Wang, H., Xia, Z., & Jiang, J. (2022). *AccMPEG: Optimizing Video Encoding for Video Analytics* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.2204.12534>
- [6] Ericsson. (2025). Mobile network traffic Q4 2025. *Ericsson Mobility Report*. <https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-update>
- [7] FFmpeg. (2025). *H.264 Video Encoding Guide* [Python]. <https://trac.ffmpeg.org/wiki/Encode/H.264>
- [8] Fraunhofer Heinrich Hertz Institute. (2021). Advanced Video Coding (H.264/AVC). *AVC*. <https://avc.hhi.fraunhofer.de/>
- [9] Galteri, L., Bertini, M., Seidenari, L., & Del Bimbo, A. (2018). Video Compression for Object Detection Algorithms. *2018 24th International Conference on Pattern Recognition (ICPR)*, 3007–3012. <https://doi.org/10.1109/ICPR.2018.8546064>
- [10] ISO/IEC. (2020). *Information technology—Coding of audio-visual objects—Part 10: Advanced video coding* (ISO/IEC 14496-10:2020; 9th ed.). International Organization for Standardization. <https://www.iso.org/standard/75400.html>
- [11] ITU-T. (2024). *H.264: Advanced video coding for generic audiovisual services* (Recommendation ITU-T H.264; Version V15). International Telecommunication Union. <https://www.itu.int/ITU-T/recommendations/rec.aspx?id=15935>
- [12] Ki, S., Bae, S.-H., Kim, M., & Ko, H. (2018). Learning-Based Just-Noticeable-Quantization- Distortion Modeling for Perceptual Video Coding. *IEEE Transactions on Image Processing*, 27(7), 3178–3193. <https://doi.org/10.1109/TIP.2018.2818439>
- [13] Kim, I.-K., Lee, S., Piao, Y., & Chen, J. (2014). Coding efficiency comparison of new video coding standards: HEVC vs VP9 vs AVS2 video. *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 1–6. <https://doi.org/10.1109/ICMEW.2014.6890700>
- [14] Kim, M.-J., & Lee, Y.-L. (2022). Object Detection-Based Video Compression. *Applied Sciences*, 12(9), 4525. <https://doi.org/10.3390/app12094525>
- [15] Li, B., Akbari, M., Liang, J., & Wang, Y. (2020). Deep Learning-Based Image Compression with Trellis Coded Quantization. *2020 Data Compression Conference (DCC)*, 13–22. <https://doi.org/10.1109/DCC47342.2020.00009>
- [16] Mao, Q., Yang, T., Zhang, Y., Wang, Z., Wang, M., Wang, S., Jin, L., & Ma, S. (2024). Extreme Image Compression Using Fine-tuned VQGANs. *2024 Data Compression Conference (DCC)*, 203–212. <https://doi.org/10.1109/DCC58796.2024.00028>
- [17] Masykuroh, K., Hendrawan, H., & Mulyana, E. (2025). Accuracy–Latency Trade-offs and Computational Complexity of Ultralytics YOLOv11 for Real-Time Video Object Detection. *2025 19th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 1–6. <https://doi.org/10.1109/TSSA68467.2025.11303782>
- [18] Metkar, S., & Talbar, S. (2013). *Motion Estimation Techniques for Digital Video Coding*. Springer India. <https://doi.org/10.1007/978-81-322-1097-9>

- [19] O'Byrne, M., Sugrue, M., Vibhoothi, & Kokaram, A. (2022). Impact of Video Compression on the Performance of Object Detection Systems for Surveillance Applications. *2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 1–8. <https://doi.org/10.1109/AVSS56176.2022.9959476>
- [20] Otani, A., Hashiguchi, R., Omi, K., Fukushima, N., & Tamaki, T. (2022). Performance Evaluation of Action Recognition Models on Low-Quality Videos. *IEEE Access*, 10, 94898–94907. <https://doi.org/10.1109/ACCESS.2022.3204755>
- [21] Reich, C., Debnath, B., Patel, D., Prangemeier, T., Cremers, D., & Chakradhar, S. (2024). Deep Video Codec Control for Vision Models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 5732–5741. <https://doi.org/10.1109/CVPRW63382.2024.00582>
- [22] Rerabek, M., Hanhart, P., Korshunov, P., & Ebrahimi, T. (2015). Quality evaluation of HEVC and VP9 video compression in real-time applications. *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, 1–6. <https://doi.org/10.1109/QoMEX.2015.7148088>
- [23] Richardson, I. E. (2011). *The H.264 Advanced Video Compression Standard*. Wiley. <https://books.google.co.id/books?id=k7nOAiUo9IC>
- [24] Rippel, O., Nair, S., Lew, C., Branson, S., Anderson, A., & Bourdev, L. (2019). Learned Video Compression. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3453–3462. <https://doi.org/10.1109/ICCV.2019.00355>
- [25] Sieber, C., Blenk, A., Hinteregger, M., & Kellerer, W. (2015). The cost of aggressive HTTP adaptive streaming: Quantifying YouTube's redundant traffic. *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 1261–1267. <https://doi.org/10.1109/INM.2015.7140478>
- [26] Sullivan, G. J., Ohm, J.-R., Han, W.-J., & Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1649–1668. <https://doi.org/10.1109/TCSVT.2012.2221191>
- [27] Uhrina, M., Frnda, J., Sevcik, L., & Vaculik, M. (2014). Impact of H.264/AVC and H.265/HEVC Compression Standards on the Video Quality for 4K Resolution. *Advances in Electrical and Electronic Engineering*, 12(4), 368–376. <https://doi.org/10.15598/aece.v12i4.1216>
- [28] Usman, M. A., Usman, M. R., & Shin, S. Y. (2017). Quality assessment for wireless capsule endoscopy videos compressed via HEVC: From diagnostic quality to visual perception. *Computers in Biology and Medicine*, 91, 112–134. <https://doi.org/10.1016/j.compbiomed.2017.10.007>
- [29] Valin, J.-M., & Terriberry, T. B. (2015). *Perceptual Vector Quantization For Video Coding*. 941009. <https://doi.org/10.1117/12.2080529>
- [30] Van Sickle, J. (2024). *FFmpeg Static Builds* [Computer software]. <https://johnvansickle.com/ffmpeg/>
- [31] Wang, Y., Wang, W., Zhang, J., Jiang, J., & Chen, K. (2019). *Bridging the Edge-Cloud Barrier for Real-time Advanced Vision Analytics*.
- [32] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), 560–576. <https://doi.org/10.1109/TCSVT.2003.815165>
- [33] Wu, F. (2014). *Advances in Visual Data Compression and Communication: Meeting the Requirements of New Applications*. CRC Press. <https://books.google.co.id/books?id=GqeNEQAAQBAJ>
- [34] Xu, J., Fookes, C., & Sridharan, S. (2018). *Automatic Event Detection for Signal-based Surveillance* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1612.01611>
- [35] Zhang, B., Jin, X., Ratnasamy, S., Wawrzyniec, J., & Lee, E. A. (2018). AWStream: Adaptive wide-area streaming analytics. *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 236–252. <https://doi.org/10.1145/3230543.3230554>
- [36] Zhong, C., & Reibman, A. R. (2018). Prediction system for activity recognition with compressed video. *Electronic Imaging*, 30(2), 254-1-254–256. <https://doi.org/10.2352/ISSN.2470-1173.2018.2.VIPC-254>