JND-aware Two-pass Per-title Encoding Scheme for Adaptive Live Streaming

Vignesh V Menon , *(Student Member, IEEE)*, Prajit T Rajendran , *(Student Member, IEEE)*,
Christian Feldmann, Klaus Schoeffmann , *(Senior Member, IEEE)*,
Mohammad Ghanbari , *(Life Fellow, IEEE)*, and Christian Timmerer , *(Senior Member, IEEE)*

***Abstract*—Adaptive live video streaming applications utilize a predefined collection of bitrate-resolution pairs, known as a *bitrate ladder*, for simplicity and efficiency, eliminating the need for additional run-time to determine the optimal pairs during the live streaming session. These applications do not incorporate two-pass encoding methods due to increased latency. However, an optimized bitrate ladder could result in lower storage and delivery costs and improved *Quality of Experience* (QoE). This paper presents a Just Noticeable Difference (JND)-aware constrained Variable Bitrate (cVBR) Two-pass Per-title encoding Scheme (`JTPS`) designed specifically for live video streaming. `JTPS` predicts a content- and JND-aware bitrate ladder using low-complexity features based on *Discrete Cosine Transform* (DCT) energy and optimizes the constant rate factor (CRF) for each representation using random forest-based models. The effectiveness of `JTPS` is demonstrated using the open source video encoder x265, with an average bitrate reduction of 18.80% and 32.59% for the same PSNR and VMAF, respectively, compared to the standard *HTTP Live Streaming* (HLS) bitrate ladder using Constant Bitrate (CBR) encoding. The implementation of JTPS also resulted in a 68.96% reduction in storage space and an 18.58% reduction in encoding time for a JND of six VMAF points.**

***Index Terms*—Per-title encoding, Live streaming, Two-pass encoding, Rate control, CRF prediction, Just Noticeable Difference.**

I. Introduction

WITH the rapid growth of online video consumption, the need for a streaming method that can adapt to varying network conditions and device capabilities became crucial. *HTTP Adaptive Streaming* (HAS) has emerged as the solution, allowing viewers to enjoy seamless playback of high-quality videos regardless of their internet connection speed or device capabilities [1]. HAS dynamically adjusts the video quality in real-time based on the viewer's context conditions (*e.g.*, network or/and device characteristics). It breaks down video content into small segments and serves them through plain HTTP [2]. Optimizing video encoding in streaming enhances the *Quality of Experience* (QoE) for the end-users and minimizes the costs for service providers, predominantly in *Video on Demand* (VoD) scenarios.

In live streaming situations where latency is crucial, video content typically employs a standardized set of encoding parameters without considering optimization. Traditionally, a fixed bitrate ladder is employed for the live streaming session, such as the *HTTP Live Streaming* (HLS) bitrate ladder[1]. However, due to the wide range of video content characteristics and network conditions, a content-adaptive approach, known as *per-title encoding* is introduced, which can improve QoE or reduce bitrate, especially for Video-on-Demand (VoD) services [3]. Although per-title encoding schemes [3]–[5] improve the quality of video delivery, they have been only appropriate for VoD streaming applications because it is computationally expensive to determine the *convex-hull*. The biggest problem in video technology today is *live (low latency)*, according to the Bitmovin Video Developer Report 2022[2]. Low-latency video coding optimization strategies are required for live-streaming applications.

Just Noticeable Difference (JND)-aware bitrate ladder prediction improves streaming by optimizing the allocation of bits based on the perceptual thresholds of human vision [6], [7]. It ensures that the available bandwidth is utilized efficiently, focusing on perceptually important areas and reducing bitrate allocation for imperceptible details [8]. This results in higher perceptual video quality within the given bitrate, enhancing the viewing experience and reducing buffering or playback interruptions [9]. Furthermore, cVBR (Constrained Variable Bitrate) encoding schemes are better than the state-of-the-art CBR (Constant Bitrate) schemes used in live streaming, owing to its ability to adapt the bitrate according to the complexity of the video content. cVBR maintains a consistent perceptual quality throughout the stream, resulting in a visually pleasing experience for viewers [10].

In this light, this paper targets a cVBR two-pass encoding scheme with a content-adaptive, JND-aware, online bitrate ladder prediction optimized for adaptive live streaming applications. The minimum and maximum encoding bitrates (b_{min}

Vignesh V Menon and Christian Timmerer are with Christian Doppler Laboratory ATHENA, Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Austria (e-mail: vignesh.menon@aau.at, christian.timmerer@aau.at).

Prajit T Rajendran is with CEA, List, F-91120 Palaiseau, Université Paris-Saclay, France (e-mail: prajit.thazhurazhikath@cea.fr).

Christian Feldmann is with Bitmovin, Austria (e-mail: christian.feldmann@bitmovin.com).

Klaus Schoeffmann is with Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Austria (e-mail: klaus.schoeffmann@aau.at).

Mohammad Ghanbari is an Emeritus Professor at the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK (e-mail: ghan@essex.ac.uk).

The financial support of the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association are gratefully acknowledged. Christian Doppler Laboratory ATHENA: https://athena.itec.aau.at/.

[1] https://developer.apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_devices, last access: May 30, 2023.

[2] https://bitmovin.com/video-developer-report, last access: May 30, 2023.

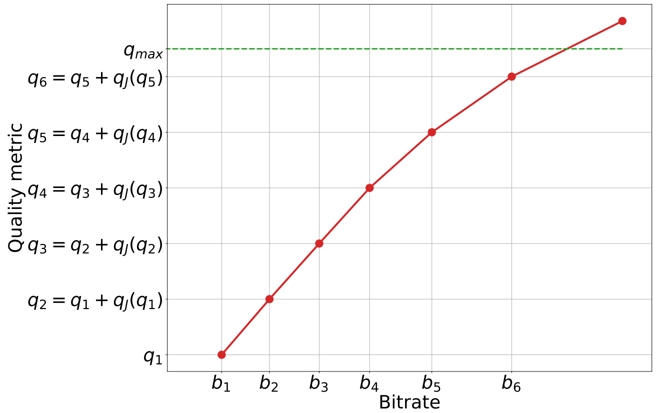


Fig. 1: The ideal JTPS bitrate ladder targeted in this paper. The red line represents the envisioned RD curve, while the green dotted line indicates the maximum quality level q_{max}. When the quality level is higher than q_{max}, the encoded video stream is considered perceptually lossless. q_J represents the target JND function.

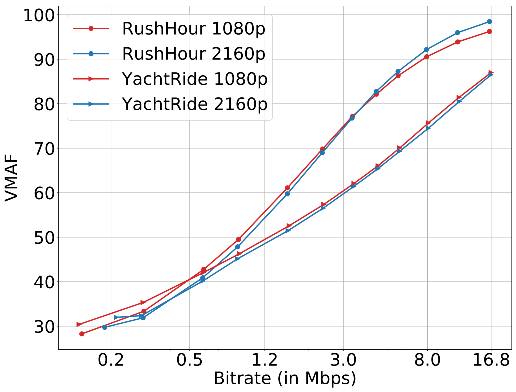


Fig. 2: Rate-Distortion (RD) curves of the Constant Bitrate (CBR) encoding of *RushHour_s000* and *YachtRide_s000* video sequences (segments) of VCD dataset [12] encoded at 1080p and 2160p resolutions using x265 HEVC encoder at *ultrafast* preset. Here, VMAF is used as the quality metric.

and b_{max}), the maximum quality level (q_{max}), and the target average JND function are considered as inputs to the scheme. Moreover, a priori information, such as the encoder/codec used and the encoding preset, are input to the scheme to ensure that the bitrate ladder is generated for the corresponding encoding configuration required by the streaming service provider. Based on the video complexity features and the input parameters, bitrate-resolution-CRF triples are predicted. As shown in Fig. 1, the adjacent points of the bitrate ladder are envisioned to have a perceptual quality difference of one JND. Please note that, in this paper, JND is considered a function of VMAF[3]. Although reducing the overall storage needed to store the representations, JTPS is expected to improve the overall compression efficiency of the bitrate ladder encoding.

In this paper, the main contributions are as follows:

① A low-latency two-pass encoding scheme termed JTPS (**J**ND-aware **T**wo-pass **P**er-Title Encoding **S**cheme) is proposed, which includes a content-adaptive, JND-aware, online bitrate ladder prediction for live video streaming applications.

② Optimized CRF is predicted for each representation for cVBR encoding to achieve the target bitrate with maximum compression efficiency.

③ Random forest-based models are designed to predict optimized bitrate-resolution-CRF triples for each video segment using *Discrete Cosine Transform* (DCT)-energy-based low-complexity spatial and temporal features of every video segment.

④ This paper also presents the extension of our previous work, OPTE [11] and PPTE [7] CBR encoding, to use random forest models using spatial and temporal features to predict perceptual quality and bitrate, instead of linear regression models. OPTE cVBR encoding scheme is introduced which includes predicting the resolution-CRF pairs for each target bitrate of the bitrate ladder, which yield the highest perceptual quality.

⑤ A comprehensive evaluation of JTPS, comparing it with state-of-the-art encoding methods, is presented.

Paper outline: Section II introduces background and related work on per-title encoding, just noticeable difference, and two-pass encoding. In Section III, the proposed scheme (JTPS) is described in detail. In Section IV, the scheme's performance is validated, and the corresponding experimental results are presented. Finally, Section V concludes the paper.

II. Background and Related Work

A. Per-title encoding

Most of the state-of-the-art per-title encoding methods is based on choosing a particular resolution that provides better visual quality for each title's bitrate range [11]. An illustration of this variation in rate-distortion (RD) characteristics can be seen in Fig. 2 for x265[4] *High Efficiency Video Coding* (HEVC) [16] encoding. For example, the cross-over bitrate between 1080p and 2160p resolutions for the *RushHour_s000* video segment occurs at around 3.4 Mbps, meaning that, for bitrates lower than 3.4 Mbps, 1080p resolution yields a higher Video Multimethod Assessment Fusion (VMAF)[5] score than 2160p. On the contrary, for bitrates higher than 3.4 Mbps, 2160p outperforms 1080p. Conversely, the *YachtRide_s000* video segment shows that 1080p resolution provides the best performance throughout the entire bitrate range, indicating that 1080p should be the resolution of choice for the entire bitrate ladder.However, The selection of bitrate-resolution pairs from the convex-hull is a challenging task. To determine the optimal per-title bitrate ladder for $\tilde{r}$ resolutions and $\tilde{b}$ bitrates, $\tilde{r} \times \tilde{b}$ test encodings are necessary. The literature describes several per-title encoding methods that reduce the number of encodings required to determine the convex-hull[6]. One such approach, developed by Katsenou *et al.* [14], uses machine

[3] Other functions can be envisioned and are subject to future work.

[4] https://www.videolan.org/developers/x265.html, last access: May 30, 2023.

[5] https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12, last access: May 30, 2023.

[6] https://bitmovin.com/per-title-encoding-datasheet/, last access: May 30, 2023.

TABLE I: Comparison of the state-of-the-art per-title encoding methods with JTPS.

Method	Target scenario	Bitrate ladder estimation method	Number of pre-encodings	Encoding type
Bruteforce [3]	VoD	Bruteforce encoding	$\tilde{r} \times \tilde{c}$	cVBR
DeepStream [13]	VoD	Bruteforce encoding	$\tilde{r} \times \tilde{c}$	cVBR
Katsenou *et al.* [14]	VoD	Bruteforce encoding	$(\tilde{r} - 1) \times 2$	CQP
FAUST [15]	VoD	Low-resolution encoding and prediction using artificial neural networks	1	CBR
Bhat *et al.* [4]	VoD	Low-resolution encoding and prediction using random forest models	1	CBR
OPTE [11]	Live	Prediction using linear regression models	0	CBR
PPTE [7]	Live	Prediction using linear regression models	0	CBR
JTPS	Live	Prediction using random forest models	0	cVBR

learning to identify the most effective bitrate range for each resolution. The method extracts spatiotemporal features and statistics from sequences at their original resolution and then, employs machine learning methods to predict the quantization parameters (QPs) at which the rate-distortion curves across the different resolutions intersect. Relatively lower number of encodes needs to be performed in order to determine the bitrates at which resolutions should be switched. This content-gnostic approach has been claimed to reduce the number of encodings required compared to other methods (by 81% - 94%) compared to the bruteforce encoding approach. Another method proposed by Bhat *et al.* [4] uses machine learning to predict the resolution without requiring multiple encodings. Features from the low resolution encoding of the first few frames are used to predict better performing resolution for a decision period. Zabrovskiy *et al.* [15] used an artificial neural network to predict an optimized bitrate ladder for each scene, optimized based on the YPSNR quality metric.

There are video encoding enhancement solutions proposed in the literature, which can be used to improve the quality of video representations (each bitrate-resolution pair) [17] in the bitrate ladder. Amirpour *et al.* [13] proposed a content-aware per-title encoding approach, *DeepStream* to support CPU-only and GPU-available end-users. However, it has limitations that: *(i)* improvements are observed only for clients with GPU, *(ii)* train deep neural networks need to be trained for each representation which needs significant processing time and *(iii)* bruteforce encoding at all resolutions and CRF are needed to estimate the bitrate ladder, making it unsuitable for real-time live streaming solutions.

Table I shows the target scenario, the bitrate estimation method, the number of pre-encodings needed to determine the convex-hull, and the encoding type of the state-of-the-art methods. The bruteforce method [3] and *DeepStream* [13] needs encoding the video content $\tilde{r} \times \tilde{c}$ times, where $\tilde{r}$ and $\tilde{c}$ denote the number of resolutions supported by the streaming service provider and number of CRFs supported by the encoder, respectively. The pre-analysis method proposed in [14] needs to encode the video $(\tilde{r}-1) \times 2$ times. Moreover, it uses constant quantization parameter (CQP) encodes which are not used in real-time streaming applications. FAUST [15] and the method proposed in [4] needs a low-resolution encoding to extract features which are input to artificial neural network and random forest models, respectively, to predict the convex-hull for CBR encoding. As a result, these methods produce *latency* significantly higher than the accepted latency in live streaming. Our previous works OPTE [11], and PPTE [7], predict optimized bitrate ladder for CBR encoding without any pre-encoding step, hence, no additional latency in streaming. They use simple linear regression models to predict bitrate-resolution pairs. There are per-title encoding methods developed in the industry: from Bitmovin[6], Brightcove [18], MUX[7], and CAMBRIA[8]. However, they are proprietary; hence, information about them is limited.

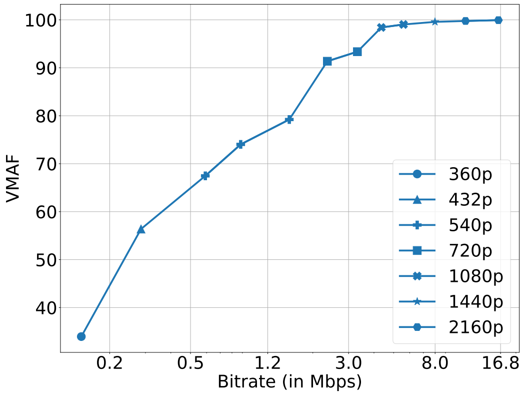


Fig. 3: RD curve of HLS[1] CBR encoding of *Characters_s000* video sequence (segment) of VCD dataset [12] using x265 HEVC encoder at *ultrafast* preset. Consequently, there is significant storage waste when these representations are stored.

B. Just Noticeable Difference (JND)

Weber's law [19] introduced the notion of Just Noticeable Difference (JND) as the change in a threshold value required to detect a difference [20]. In visual perception, JND refers to the slightest distinguishable difference between two levels of sensory stimulus [21]. Additionally, JND represents the maximum tolerable level of distortion for the *Human Visual System* (HVS) when perceiving videos. Research has been conducted on JND, and several surveys have been published [8], [22]–[26]. By utilizing JND in video coding, referred to as *perceptual coding*, the encoding bitrate can be reduced while still guaranteeing a certain level of video quality or minimizing distortion within a specific bitrate constraint. Furthermore, removing perceptual redundancy information from JND levels compared to traditional video coding methods can lead to additional compression gains [6]. For instance, Fig. 3 shows

[7] https://www.mux.com/blog/instant-per-title-encoding, last access: May 30, 2023

[8] https://capellasystems.net/wp-content/uploads/2021/01/CambriaFTC_SABL.pdf, last access: May 30, 2023

the selected bitrate-resolution pairs and their VMAF[5] scores for the *Characters_s000* sequence using the HLS bitrate ladder. It is seen that there are multiple representations with the same video quality and some with similar quality at mid-range bitrates. Choosing representations with similar quality does not enhance QoE, but it increases storage and bandwidth expenses [9].

C. Two-pass encoding

Most streaming service providers employ CBR rate-control mode to encode live videos. CBR's consistency in achieved bitrate makes it more dependable for time-sensitive data delivery, especially in live streaming applications. Bitrate overshoot, or the encoded bitrate exceeding internet speeds, is not a concern because CBR-encoded videos are streamed consistently. However, this method's dependability sometimes necessitates sacrificing compression efficiency[9]. In contrast, VoD applications utilize Variable Bitrate (VBR), where video segments are encoded according to their content complexity to optimize the transmission at the expense of adding a pre-processing stage to evaluate the content complexity of the video segments (*two-pass encoding*). As shown in Fig. 4, the input data from the video is analyzed (and stored in a log file) in the first-pass of two-pass encoding. The collected data from the first-pass is used to achieve the best encoding compression efficiency in the second-pass. During the second-pass encoding, bitrate is allocated among segments based on content complexity such that the average bitrate remains constant. This fluctuating characteristic makes VBR best suited for VoD applications [10].

Two-pass encoding had been the *de-facto* solution proposed to distribute bits effectively and improve the compression efficiency in VoD applications. Other than the previously discussed pre-analysis methods, some schemes involve encoding the same content twice to adapt the encoding parameters *per title*. Que *et al.* [27] proposed a two-pass VBR method for *Advanced Video Coding* (AVC) [28]. The first-pass uses CBR encoding to gather encoding statistics, while offline processing is used in the second-pass to detect scene-cuts, precisely allocate target bits, and determine the quantization parameter for each frame. Zupancic *et al.* [29] utilized a fast encoder with a condensed set of coding tools in the first-pass to collect data for rate allocation and model parameter initialization during the second-pass. Wang *et al.* [30] proposed a two-pass VBR control for HEVC, motivated by structural similarity (SSIM), that allocates available bits at the group of pictures (GOP), frame, and coding unit (CU) levels to create a perceptually uniform space. Since the two-pass encoding method generally involves processing all segments twice, the overall encoding time is increased two-fold, introducing added streaming latency. Hence, these schemes are not used for live video streaming.

Constrained Variable Bitrate (cVBR) is the most widely used type of two-pass Variable Bitrate encoding[9] [10]. This encoding scheme involves setting a maximum bitrate and buffer window, requiring two encoding passes to complete the process. The target bitrate cannot be specified in Constant Rate Factor (CRF) rate control mode, so the information from the first-pass is used to determine the optimized CRF that achieves the target bitrate. During the second-pass, the video segment is encoded with the selected optimized CRF while maintaining the maximum bitrate and buffer window constraints. This results in reaching the desired target bitrate with maximum compression efficiency[10]. In terms of computational costs, CBR encoding generally incurs lower computational costs due to its fixed and predictable bitrate allocation. CVBR encoding, with its complexity analysis and bitrate adjustments, requires more computational resources. However, the specific computational cost can vary depending on factors such as video resolution, content complexity, and hardware capabilities.

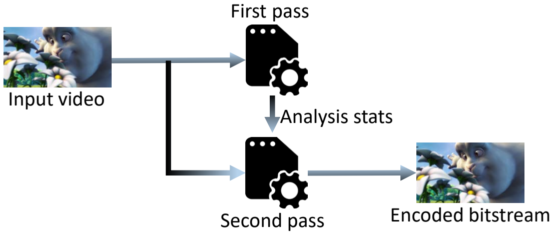


Fig. 4: Two-pass encoding architecture.

Another popular method of two-pass encoding is to extract video complexity features as the first-pass, and use them to predict encoding parameters in the second-pass. Low-complexity features must be chosen in live streaming applications to guarantee uninterrupted low-latency video streaming. An intuitive method for feature extraction would be to utilize Convolutional Neural Networks (CNNs). However, CNN-based feature extraction would not be effective as it lacks temporal motion information, which is crucial for video complexity detection and subsequent bitrate-ladder prediction. Architectures such as 3D-CNN [31] or Conv-LSTM [32], [33] could be alternatives to accommodate the temporal motion information present in the video stream. However, such models have several inherent disadvantages, such as higher training time, inference time, and storage requirements (to deploy the prediction models in real-time), which are impractical in live streaming applications. Although CNN-based approaches could result in rich features, simpler models which yield a significant prediction performance are more suitable for live video streaming. The popular state-of-the-art video complexity features are Spatial Information (SI) and Temporal Information (TI) [34]. The rate of SI and TI feature extraction[11] from 2160p resolution videos are observed as around five frames per second, which is insufficient for low-latency streaming applications [35].

To summarize, most related works on per-title and two-pass encoding yield latency unsuitable for live-streaming applications. Machine learning-based methods in the literature are too complex and storage heavy, hence, they are not suitable

[9] https://docs.aws.amazon.com/mediaconvert/latest/ug/mediaconvert-guide.pdf, last access: May 30, 2023.

[10] https://www.wowza.com/blog/cbr-vs-vbr, last access: May 30, 2023

[11] https:// github.com/Telecommunication-Telemedia-Assessment/SITI, last access: May 30, 2023.

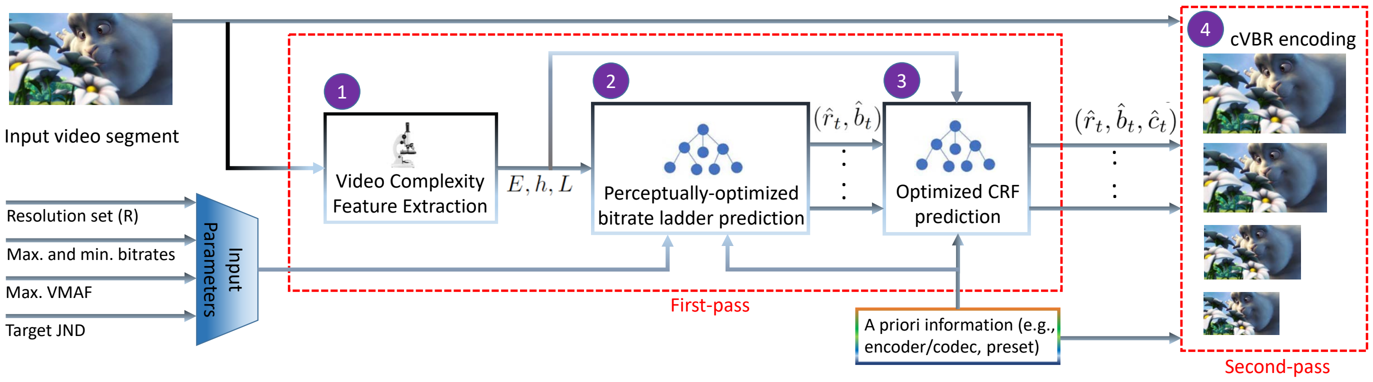


Fig. 5: Live HTTP adaptive streaming featuring our JND-aware two-pass per-title encoding scheme (JTPS).

for real-time deployments. To overcome these problems, this paper proposes a low-latency pre-processing step as the first-pass to analyze the video segment's complexity to predict an optimized encoding bitrate-ladder.

III. JND-aware Two-pass Per-title Encoding Scheme (JTPS)

The architecture of the proposed JTPS scheme for live video streaming applications is shown in Fig. 5. For each segment of the input video sequence, the JND-aware bitrate ladder is determined so that the adjacent RD points of the bitrate ladder have a perceptual quality difference of one JND. The prediction of every segment is motivated by the fairly uniform frame-to-frame spatiotemporal content of frames within a segment [1]. The bitrate ladder is predicted using video complexity features (*i.e.*, E, h, and L features are explained in Section III-A) extracted for every segment and the set of pre-defined resolutions (R), minimum and maximum target bitrates (*i.e.*, b_{min} and b_{max}), average JND quality (v_J) function, and the maximum VMAF (v_{max}) of the bitrate ladder. This paper assumes that VMAF is the optimal measure of perceptual quality[12]. To ensure that the predicted bitrates and VMAF values align with the preferences of the streaming service provider, JTPS takes inputs b_{min}, b_{max}, and v_{max}. By considering b_{max} and v_{max}, JTPS can be adjusted to optimize the number of representations in the bitrate ladder. Additionally, the input R ensures that only the supported resolutions of the streaming service provider are selected for the encoding set. The process starts by predicting the VMAF corresponding to b_{min}. The VMAF scores for the remaining representations are calculated by incrementing the previous VMAF in the bitrate ladder by one JND until either b_{max} or v_{max} is reached. These VMAF values are then used to predict the corresponding bitrate-resolution pairs. Additionally, an optimized CRF is predicted to achieve maximum compression efficiency for the cVBR encoding of the selected bitrate-resolution pairs. For each segment, the encoding process is performed exclusively for the predicted perceptually aware bitrate-resolution-CRF triples. In this manner, compression efficiency is improved over traditional fixed bitrate ladder and CBR encoding schemes while decreasing storage and, consequently, content delivery network (CDN) costs. JTPS is classified into four phases (cf. Fig. 5; the first-pass comprises the first three phases and the second-pass comprises the last/forth phase):

① Video complexity feature extraction (Section III-A)
② Perceptually-optimized bitrate ladder prediction (Section III-B)
③ Optimized CRF prediction for the selected bitrate-resolution pairs (Section III-C)
④ cVBR encoding of the segments using the predicted bitrate-resolution-CRF triples

Optimized bitrate prediction and CRF prediction are separated into two distinct prediction modules for better interpretability and control over the prediction process. The first module derives the target resolution ($\hat{r}_t$) and the upper limit for the instantaneous bitrate ($\hat{b}_t$), while the second module derives the CRF parameter ($\hat{c}_t$) based on ($\hat{r}_t$, $\hat{b}_t$). Utilizing a two-module approach is advantageous, as it explicitly helps us model and optimize for different aspects of the problem.

A. Video Complexity Feature Extraction

In this paper, three DCT-energy-based features, *(i)* the average texture energy E, *(ii)* the average gradient of the texture energy h, and *(iii)* the average luminescence L are used as the spatial and temporal complexity measures [11], [35]. The feature extraction method was proposed in our previous work [35] and is included here to have the paper self-contained.

The following DCT-based energy function is used to determine the texture of every non-overlapping block k in each frame f, which is defined as:

$$H_{f,k} = \sum_{i=0}^{w-1}\sum_{j=0}^{w-1} e^{|(\frac{ij}{w^2})^2-1|}|DCT(i,j)| \tag{1}$$

where $w \times w$ pixels is the size of the block, and $DCT(i,j)$ is the $(i,j)^{th}$ DCT component when $i+j > 0$, and 0 otherwise [36]. To determine the *spatial energy* feature per segment, denoted as E, the texture is averaged as illustrated below:

$$E = \sum_{f=0}^{F-1}\sum_{k=0}^{K-1} \frac{H_{f,k}}{F \cdot K \cdot w^2} \tag{2}$$

[12] Other quality metrics can be envisioned and are subject to future work.

Algorithm 1: Bitrate ladder prediction algorithm

```
Inputs:
  $b_{min}, b_{max}$ : minimum and maximum target bitrate
  $R$ : set of all resolutions $r$
  $v_J$ : average (target) JND function
Output: $(\hat{r}, \hat{b})$ pairs of the bitrate ladder
Step 1: $\hat{b}_1 = b_{min}$
  Determine $\hat{v}_{r,\hat{b}_1}$ $\forall r \in R$
  $\hat{v}_1 = max(\hat{v}_{r,\hat{b}_1})$
  $\hat{r}_1 = \arg\max_{r \in R}(\hat{v}_{r,\hat{b}_1})$
  $(\hat{r}_1, \hat{b}_1)$ is the first point of the bitrate ladder.
Step 2:
  $t = 2$
  while $\hat{b}_{t-1} < b_{max}$ and $\hat{v}_{t-1} < v_{max}$ do
    $\hat{v}_t = \hat{v}_{t-1} + v_J(\hat{v}_{t-1})$
    Determine $\hat{b}_{r,\hat{v}_t}$ $\forall r \in R$
    $\hat{b}_t = min(\hat{b}_{r,\hat{v}_t})$
    $\hat{r}_t = \arg\min_{r \in R}(\hat{b}_{r,\hat{v}_t})$
    $(\hat{r}_t, \hat{b}_t)$ is the $t^{th}$ point of the bitrate ladder.
    $t = t + 1$
```

Here, K represents the number of blocks per frame, and F denotes the number of frames per segment. Furthermore, the block-wise sum of absolute difference (SAD) of the texture energy of each frame compared to its previous frame is computed and then averaged per segment to obtain the average *temporal energy* (h) as shown below:

$$h = \sum_{f=1}^{F-1} \sum_{k=0}^{K-1} \frac{| H_{f,k} - H_{f-1,k} |}{(F-1) \cdot K \cdot w^2} \tag{3}$$

The luminescence of non-overlapping blocks k of each frame p is defined as:

$$L_{f,k} = \sqrt{DCT(0,0)} \tag{4}$$

where $DCT(0,0)$ is the DC component in the DCT calculation. The block-wise luminescence is averaged per segment denoted as L, as shown below.

$$L = \sum_{f=0}^{F-1} \sum_{k=0}^{K-1} \frac{L_{f,k}}{F \cdot K \cdot w^2} \tag{5}$$

Please note that E and L represent the spatial characteristics of the video segment, while h represents the temporal characteristic, which are used in the following steps to predict the encoding bitrate ladder.

B. Perceptually-optimized Bitrate Ladder Prediction

The JND-aware bitrate ladder prediction method is presented in Algorithm 1 and comprises two steps.

Step 1: The perceptual quality, measured by VMAF, is modeled as a function of features such as E, h, and L, resolution r, and target bitrate b, which can be expressed as $v_{r,b} = f(E, h, L, r, b)$ [37]. The first point in the bitrate ladder is determined by predicting VMAF for all resolutions $r \in R$ at $\hat{b}_1 = b_{min}$ (as shown in Fig. 6) using VMAF prediction models. From the predicted VMAF values (*i.e.*, $\hat{v}_{r,\hat{b}_1}$ values) for different resolutions, the resolution with the highest VMAF value, $\hat{r}_1$, is selected to correspond to the bitrate $\hat{b}_1$. This results in the first point of the bitrate ladder being $(\hat{r}_1, \hat{b}_1)$.

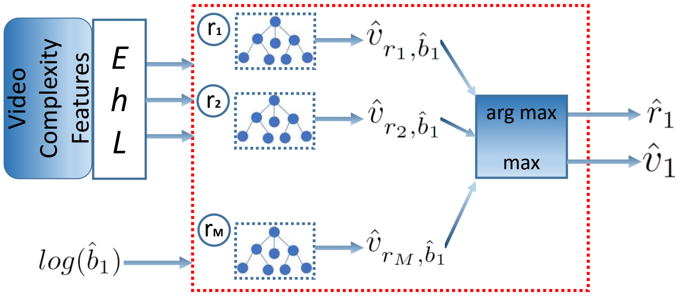

Video Complexity Features
E
h
L
$log(\hat{b}_1)$
r₁
r₂
r_M
$\hat{v}_{r_1,\hat{b}_1}$
$\hat{v}_{r_2,\hat{b}_1}$
$\hat{v}_{r_M,\hat{b}_1}$
arg max
max
$\hat{r}_1$
$\hat{v}_1$

Fig. 6: Estimation of the first point of the bitrate ladder. $\hat{v}_1$ is the maximum value among the $\hat{v}_{r,\hat{b}_1}$ values output from the predicted models trained for resolutions r_1, .., r_M. The resolution corresponding to the VMAF $\hat{v}_1$ is chosen as $\hat{r}_1$.

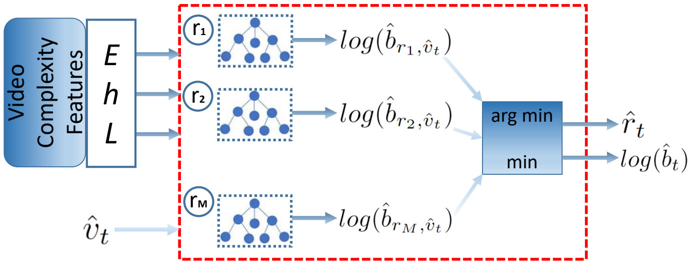

Video Complexity Features
E
h
L
$\hat{v}_t$
r₁
r₂
r_M
$log(\hat{b}_{r_1,\hat{v}_t})$
$log(\hat{b}_{r_2,\hat{v}_t})$
$log(\hat{b}_{r_M,\hat{v}_t})$
arg min
min
$\hat{r}_t$
$log(\hat{b}_t)$

Fig. 7: Estimation of the t^{th} point ($t > 1$) of the bitrate ladder. $log(\hat{b}_t)$ is the minimum value among the $log(\hat{b}_{r,\hat{v}_t})$ values output from the predicted models trained for resolutions r_1, .., r_M. The resolution corresponding to $log(\hat{b}_t)$ is chosen as $\hat{r}_t$.

Step 2: For every subsequent point in the bitrate ladder ($t > 1$), the target VMAF is set to $\hat{v}_t = \hat{v}_{t-1} + v_J(\hat{v}_{t-1})$, which means one JND more than the previous point. Bitrate is modeled as a function of E, h, L features, resolution r, and target VMAF v, *i.e.*, $b_{r,v} = f(E, h, L, r, v)$. The target bitrate $\hat{b}_{r,\hat{v}_t}$ required to achieve the VMAF $\hat{v}_t$ is determined for each resolution in R (refer to Fig. 7). The minimum value of $\hat{b}_{r,\hat{v}_t}$ in all resolutions is considered as $\hat{b}_t$ for the bitrate ladder, and the resolution corresponding to the minimum value is chosen as $\hat{r}_t$. This process is repeated until $\hat{b}_t$ is greater than or equal to b_{max} or $\hat{v}_t$ is greater than or equal to v_{max}.

Implementation of prediction models: The prediction models are trained for each resolution supported by the streaming service provider, ensuring the scalability of the design without the need to retrain the entire network when adding a new resolution to the framework. In this paper, the following prediction models *(i)* linear regression model [38], *(ii)* XGBoost[13] [39] and *(iii)* random forest regression model[14] [40], are used and compared for their prediction accuracy in terms of R^2 score and Mean Absolute Error (MAE). Random forest regressor is

[13] https://xgboost.readthedocs.io/en/stable/parameter.html, last access: May 30, 2023.

[14] https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees, last access: May 30, 2023.

TABLE II: Prediction accuracy of VMAF and $log(b)$ prediction models for 2160p resolution encoding of VCD dataset [12] using x265 HEVC encoder at *ultrafast* preset.

	VMAF prediction		$log(b)$ prediction	
Method	R^2	MAE	R^2	MAE
Linear Regression	0.598	10.102	0.651	1.126
XGBoost	0.877	4.484	0.896	0.511
Random Forest	**0.930**	**3.941**	**0.943**	**0.488**

an ensemble regression model that uses a randomly selected subset of training samples and variables to train multiple decision trees in parallel, commonly known as bagging. The cumulative results of all the numerous decision trees in the ensemble are combined to obtain the final predictions.

Table II shows the results of the VMAF and $log(b)$ prediction, respectively, using the models mentioned above for 2160p resolution using the default hyper-parameters of the models[13,14]. It is observed that the R^2 score is the maximum and the MAE score is the minimum for random forest models. Moreover, random forest models exhibit lower overfitting than XGB and are faster as the decision trees run in parallel (courtesy of distributed computing-based approaches). Hence, this paper uses random forest models for VMAF and $log(b)$ prediction for each resolution. Please note that training models for each resolution ensure scalability, as more resolutions can be added to `JTPS` architecture in the future with minimal retraining. Hyper-parameter tuning is performed on the prediction models of 2160p to obtain a balance between model size and performance. The selected hyper-parameters[14] for VMAF and $log(b)$ prediction models are *min_samples_leaf*=1, *min_samples_split*=2, *n_estimators*= 00, and *max_depth*=14.

The total processing time of the bitrate ladder prediction algorithm (τ_B) is:

$$\tau_B = (\tilde{r} \cdot \tau_{vp}) + (N - 1) \cdot (\tilde{r} \cdot \tau_{bp}) \tag{6}$$

where $\tilde{r}$ and N denote the number of resolutions in R and the number of points in the bitrate ladder, respectively. τ_{vp} denotes the inference time of the VMAF prediction models and τ_{bp} represents the inference time of the bitrate prediction models. The amount of memory required to store the models for bitrate ladder prediction (s_B) is given by:

$$s_B = \sum_{r=1}^{\tilde{r}} (s_{vp_r} + s_{bp_r}) \tag{7}$$

where s_{vp_r} denotes the size of the VMAF prediction model trained for the resolution r, and s_{bp_r} denotes the size of the bitrate prediction model trained for the resolution r.

C. Optimized CRF Prediction

For HAS it is essential to avoid exceeding the maximum bitrates specified in the HLS/DASH manifests [2] during the encoding process. Failure to adhere to these limits can lead to buffer overflows or underflows in video players[10]. Therefore, accurately predicting CRF becomes of utmost importance. In this paper, CRF is predicted instead of quantization parameter (QP), since, it simplifies the encoding workflow by eliminating the need to manually set and adjust QP for each frame. Once the bitrate ladder is determined, the optimized CRF $\hat{c}_t$ is estimated for every ($\hat{r}_t$, $\hat{b}_t$). CRF c is modeled as a function of the features E, h, and L, the resolution r, and the target bitrate b, *i.e.*, $c_{r,b} = f(E, h, L, r, b)$. A prediction model is trained for each resolution r, which determines $\hat{c}_t$ based on E, h, L, and $log(\hat{b}_t)$ for each video segment as shown in Fig. 8. The minimum and maximum CRF (c_{min} and c_{max} respectively) are chosen based on the target video encoder. For example, x264[15] AVC [28] encoder and x265[4] HEVC [16] encoder support a CRF range between 0 and 51. SVT-AV1[16], an AV1 [41] encoder supports a CRF range between 1 and 63.

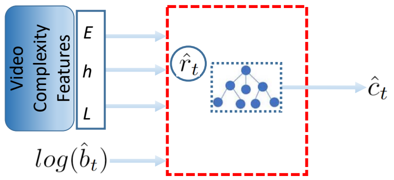


Fig. 8: Estimation of the optimized CRF to achieve the target bitrate $\hat{b}_t$ using a prediction model trained for resolution $\hat{r}_t$.

TABLE III: Prediction accuracy of CRF prediction models for 2160p resolution encoding of VCD dataset [12] using x265 HEVC encoder at *ultrafast* preset.

Method	R^2	MAE
Linear Regression	0.873	3.976
XGBoost	0.961	2.203
Random forest	**0.968**	**1.871**

Implementation of prediction models: Similarily as for the bitrate prediction, linear regression model [38], XGBoost [39], and random forest regression model [40] are tested for their prediction accuracy in terms of R^2 score and MAE. As shown in Table III, R^2 score is the maximum, and the MAE score is the minimum for random forest models. Furthermore, random forest models for CRF prediction for every resolution exhibit a lower tendency of over-fitting and can utilize distributed computing for faster training and prediction. Hyper-parameter tuning is performed on the prediction model of 2160p to obtain a balance between model size and performance. The selected hyper-parameters[14] are *min_samples_leaf*=1, *min_samples_split*=2, *n_estimators*=100, and *max_depth*=14. Since the output of the prediction model is a floating point value, the decimal value is truncated so that the result is an integer.

The total processing time of the CRF prediction (τ_C) is:

$$\tau_c = N \cdot \tau_{cp} \tag{8}$$

where τ_{cp} denotes the inference time of the CRF prediction models. The amount of memory required to store the models for CRF prediction (s_C) is given by:

$$s_c = \sum_{r=1}^{\tilde{r}} (s_{cp_r}) \tag{9}$$

[15] https://www.videolan.org/developers/x264.html, last access: May 30, 2023.

[16] https://github.com/AOMediaCodec/SVT-AV1, last access: May 30, 2023.

where s_{vp_r} denotes the size of the CRF prediction model trained for resolution r.

IV. Evaluation

A. Test Methodology

The Video Complexity Dataset [12] is used to validate the performance of the encoding schemes considered in this paper. The dataset needed to train and test the prediction models is generated as shown in Algorithm 2. E, h, and L features are extracted using VCA v2.0[17] open-source video complexity analyzer [35]. The sequences are encoded at 30 fps using x265 v3.5[4] with the *ultrafast* preset on a dual-processor *encoding server* with Intel Xeon Gold 5218R (80 cores, frequency at 2.10 GHz). VCA and x265 are run using a single thread with only x86 SIMD optimization [42] to compare the time complexity of the considered schemes. The resolutions specified in Apple HLS authoring specifications[1] are considered in the evaluation, *i.e.*, R= {360p, 432p, 540p, 720p, 1080p, 1440p, 2160p}. The total memory used to store VMAF and bitrate prediction models for bitrate ladder prediction, s_B (*cf.* Eq. 7) is 777 MB, *i.e.*, 58 MB for VMAF prediction models for each resolution, and 53 MB for bitrate prediction models for each resolution. The total memory used to store CRF prediction models, s_c (*cf.* Eq. 9) is 400 MB, *i.e.*, 57 MB for CRF prediction models for each resolution. In order to check for the generalization of the models, 5-fold cross-validation is performed, and the values are averaged from all folds. Since these values are similar, we assume that the model generalizes well. It was ensured that the training set does not include any segments from the same scenes in the test set. For a target bitrate of b_t (in Mbps), the CBR encoding is achieved by setting the *bitrate* and *vbv-maxrate*[18] option of x265 as b_t, and enabling *strict-cbr* flag[18]. Similarly, for a target bitrate of b_t (in Mbps) and CRF c_t, cVBR encoding is achieved by setting the *crf* option[18] of x265 as c_t, and *vbv-maxrate* option as b_t.

This paper considers the following encoding schemes to compare with JTPS:

- Bruteforce bitrate ladder encoding [3], where the bitrate-resolution-CRF triples are determined by encoding videos using all CRFs supported by x265 for all resolutions. The representations are chosen such that there is a VMAF difference of one target JND.
- HLS CBR encoding, which is the CBR encoding of HLS bitrate ladder[1].
- OPTE [11] CBR encoding, where optimized resolution is predicted for the set of bitrates in the HLS bitrate ladder, as shown in Fig. 6. In [11], linear regression models were used to predict VMAF based on the E and h features. For the evaluation in this paper, the method is extended by using random forest models trained to predict VMAF (for all resolutions in R) based on the E, h, and L features using the CBR encoding dataset (*cf.* Algorithm 2).

Algorithm 2: Dataset generation.

```
cVBR encoding dataset
Inputs:
  $R$: set of resolutions
  $c_{min}$: minimum supported CRF
  $c_{max}$: maximum supported CRF
for each video segment do
    Determine $E$, $h$, and $L$
    for each $r \in R$ do
        for each $c \in [c_{min}, c_{max}]$ do
            Encode segment with CRF $c$ ;
            Record $E$, $h$, $L$, $r$, $c$, achieved bitrate $b'$,
              VMAF $v$, and PSNR $p$ ;
CBR encoding dataset
Inputs:
  $R$: set of resolutions
  $B$: set of target bitrates
for each video segment do
    Determine $E$, $h$, and $L$
    for each $r \in R$ do
        for each target bitrate $b \in B$ do
            Encode segment with CBR $b$ ;
            Record $E$, $h$, $L$, $r$, $b$, achieved bitrate $b'$,
              VMAF $v$, and PSNR $p$ ;
```

- PPTE [7], where optimized bitrate-resolution pairs are predicted for JND-aware CBR encoding as shown in Algorithm 1. In [7], linear regression models were used to predict VMAF and bitrate based on the E and h features. For the evaluation in this paper, the method is extended by using random forest models trained to predict VMAF and bitrate (for all resolutions in R) based on the E, h, and L features using the CBR encoding dataset (*cf.* Algorithm 2).
- HLS cVBR encoding, where the optimized CRF is predicted for the bitrate-resolution pairs of the HLS bitrate ladder, as shown in Fig. 8. Random forest models are trained to predict CRF (for all resolutions in R) using the cVBR encoding dataset (*cf.* Algorithm 2).
- OPTE cVBR encoding, where the optimized CRF is predicted along with the optimized resolution for the set of bitrates in the HLS bitrate ladder for cVBR encoding. This scheme predicts VMAF for all resolutions in R for a given set of target bitrates. The resolution which yields the maximum VMAF is chosen as the optimized resolution for the given target bitrate, as shown in Fig. 6. Random forest models are trained to predict VMAF and CRF (for all resolutions in R) using the cVBR encoding dataset (*cf.* Algorithm 2).

For PPTE and JTPS encoding, the parameters, b_{min}, and b_{max} are set as 0.145 Mbps and 16.8 Mbps, respectively, to compare with the HLS bitrate ladder. The average target JND function (v_J) is considered as two [26], four, and six[19] based

[17] https://vca.itec.aau.at, last access: May 30, 2023.

[18] https://x265.readthedocs.io/en/master/cli.html#quality-rate-control-and-rate-distortion-options, last access: May 30, 2023.

[19] https://streaminglearningcenter.com/codecs/finding-the-just-noticeable-difference-with-netflix-vmaf.html, last access: May 30, 2023.

on current industry practices. Accordingly, v_{max} is set as 98, 96, and 94, respectively, to comply with the target JND value.

First, the pre-processing time (τ_p), *i.e.*, latency in encoding due to the time taken for video complexity feature extraction, and the inference time of the models to predict the optimized bitrate-resolution-CRF triples are determined to evaluate the first-pass encoding time. τ_p for state-of-the-art methods [3], [13]–[15], [43] is the time for pre-encoding. The additional computational time overhead to determine convex-hull ΔT_C is reported as a ratio to the sum of encoding times of all representations in the reference bitrate ladder encoding as shown below:

$$\Delta T_C = \frac{\tau p}{\sum t_{ref}} \tag{10}$$

Second, the VMAF, $log(b)$, and CRF prediction models are assessed in terms of the prediction accuracy using the coefficient of determination (R^2) score and Mean Absolute Error (MAE) compared to the ground truth values. The achieved VMAF, bitrate, and CRF recorded in the cVBR encoding dataset are ground truth values. Third, the relative importance of the features used is evaluated using the SHapley Additive exPlanations (SHAP) values [44].

The encoding schemes' rate-distortion (RD) curves are analyzed for selected video sequences (segments) of various video content complexities. Bjøntegaard delta rates [45] BDR_P and BDR_V refer to the average increase in bitrate of the representations compared with that of the reference bitrate ladder encoding scheme to maintain the same PSNR and VMAF, respectively. A negative BDR suggests a boost in the coding efficiency of the considered encoding scheme compared to the reference bitrate ladder encoding scheme. BD-PSNR and BD-VMAF refer to the average increase in PSNR and VMAF, respectively, at the same bitrate compared with the reference bitrate ladder encoding scheme. A positive BD-PSNR and BD-VMAF denote an increase in the coding efficiency of the considered encoding scheme compared to the reference bitrate ladder encoding scheme.

The relative difference in the storage space needed to store all bitrate ladder representations of the considered encoding scheme (ΔS) is also evaluated as:

$$\Delta S = \frac{\sum b_{opt}}{\sum b_{ref}} - 1 \tag{11}$$

where $\sum b_{ref}$ and $\sum b_{opt}$ represent the sum of bitrates of all representations in the reference bitrate ladder encoding and the bitrate ladder encoding using the considered encoding scheme, respectively. Similarly, the relative difference in the encoding time of the considered encoding scheme (ΔT) is also evaluated as:

$$\Delta T = \frac{\tau_p + \sum t_{opt}}{\sum t_{ref}} - 1 \tag{12}$$

where $\sum t_{ref}$ and $\sum t_{opt}$ represent the sum of encoding times of all representations in the reference bitrate ladder encoding and the bitrate ladder encoding using the considered encoding scheme, respectively.

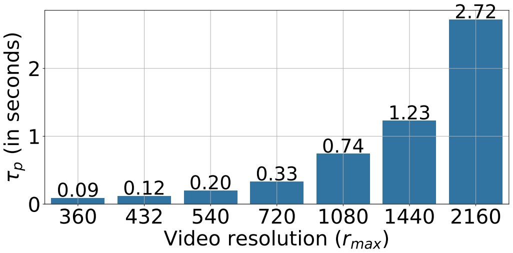


Fig. 9: Pre-processing time (τ_p) of JTPS for various input video resolutions.

TABLE IV: Comparison of the additional computational time overhead to determine the convex-hull.

Method	ΔT_C
Bruteforce [3]	4596.77%
DeepStream [13]	4596.77%
Katsenou *et al.* [14]	120.57%
FAUST [15]	48.65%
Bhat *et al.* [4]	67.82%
OPTE [11]	0.30%
PPTE [7]	0.33%
JTPS	0.41%

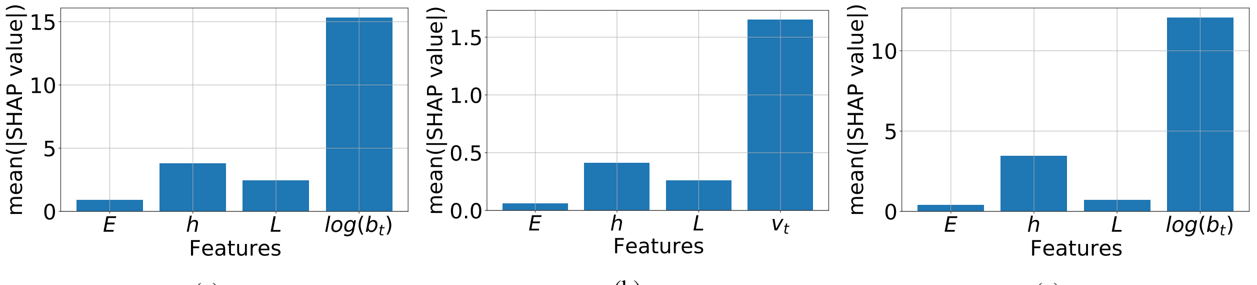
B. Experimental Results

This section presents the results of JTPS. The pre-processing time (τ_p), *i.e.*, the sum of feature extraction time and the inference time of the prediction models is evaluated. E, h, and L features are extracted at an average speed of 44 frames per second over the entire dataset, *i.e.*, for a segment of four second duration, features are extracted in 2.71 s. The average inference time of the random forest models for the bitrate ladder and CRF prediction (τ_{vp}, τ_{bp}, and τ_{cp}) is 5 ms. Hence, τ_p is 2.72 s. As observed in Fig. 9, τ_p decreases as the video resolution (r_{max}) decreases. The inference time of the prediction models do not change, however, the featrue extraction time reduces considerably as the resolution decreases. In real-time applications, video complexity feature extraction and the encoding bitrate-ladder prediction can be executed as concurrent processes, using multi-threading optimizations. As an example, τ_p is reduced to 0.35 s when eight CPU threads are used for feature extraction. As shown in Table I, the state-of-the-art methods have pre-encoding steps to determine convex-hull, making them unsuitable for live streaming applications. However, OPTE [11], PPTE [7] and JTPS do not need pre-encoding. Table IV shows the additional computational time overhead needed to determine the convex-hull (first-pass encoding time) compared to HLS CBR encoding time. It is observed that our previous works OPTE and PPTE, and JTPS need significantly lower processing time to predict the bitrate ladder, compared to the state-of-the-art methods; hence, they are suitable for live streaming applications.

The performance of the VMAF, bitrate and CRF prediction models are investigated using the R^2 score and MAE, as shown in Table V. The average R^2 score of the VMAF, bitrate, and CRF prediction models are estimated as 0.886, 0.910, and 0.968, respectively. Hence, it can be observed that there is a strong positive correlation between the predicted and ground

TABLE V: R^2 score and MAE of the prediction models for various resolutions.

	R^2 score							MAE						
r	360p	432p	540p	720p	1080p	1440p	2160p	360p	432p	540p	720p	1080p	1440p	2160p
VMAF	0.821	0.852	0.882	0.906	0.910	0.906	0.930	5.091	5.071	4.966	4.971	4.806	4.490	3.941
log(b)	0.867	0.884	0.901	0.910	0.932	0.937	0.943	0.527	0.505	0.472	0.456	0.460	0.472	0.488
CRF	0.969	0.969	0.970	0.969	0.968	0.967	0.968	1.823	1.820	1.820	1.859	1.860	1.885	1.871



(a) (b) (c)

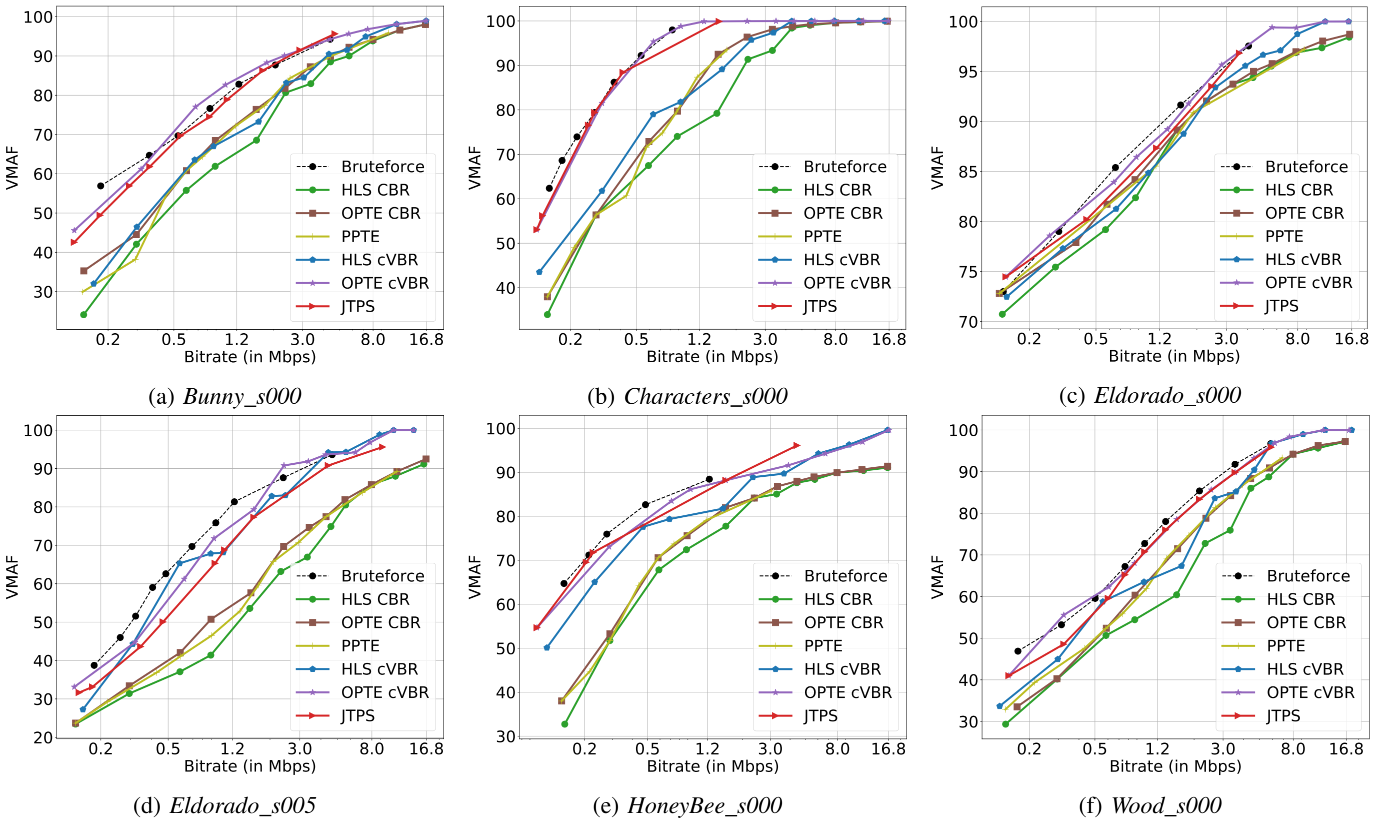
Fig. 10: The relative importance of (a) VMAF prediction, (b) bitrate prediction, and (c) CRF prediction for 2160p resolution determined by SHAP values [44].

truth values. The average MAE of the prediction models is estimated as 4.762, 0.483, and 1.848, respectively, which is acceptable in live streaming applications. Furthermore, this paper also examines the relative feature importance in the prediction models. Fig. 10 shows the SHAP values [44] corresponding to the features used in the prediction models. The target bitrate in the logarithmic scale ($log(b_t)$) is the most influential feature for VMAF prediction, followed by the h, L, and E features. Similarly, target VMAF (v_t) is the most important feature for bitrate prediction, followed by the h, L, and E features. Furthermore, $log(b_t)$ is the most vital feature for CRF prediction, followed by the h, L, and E features. Intuitively, lower CRF yields higher bitrate and VMAF, and vice versa. Additionally, in inter-coding, temporal activity is expected to influence the encoding decisions more than spatial content. Hence, h is expected to be more critical in the predictions than L and E, respectively.

Fig. 11 shows the RD curves of selected video sequences (segments) of various video complexities with bruteforce encoding [3], HLS CBR encoding, `OPTE` CBR encoding [11], `PPTE` encoding [7], HLS cVBR encoding, `OPTE` cVBR encoding, and `JTPS`. It is observed that `JTPS` determines the RD points so that the average VMAF difference between consecutive RD points is the target JND value (in the figure, JND is assumed as 6 VMAF points). Furthermore, the VMAF achieved by `JTPS` is higher than HLS CBR encoding at the same target bitrates. In most cases, however, `OPTE` cVBR yields higher VMAF than the other encoding schemes at the same target bitrates for videos in all complexity classes. This is because `OPTE` cVBR encoding is optimized for maximizing VMAF, while `JTPS` is a joint optimization for maximizing VMAF and maintaining a perceptual gap between representations. Hence, the number of representations in `JTPS` for every video segment is lower than for HLS ladders and `OPTE` encoding. On average, `JTPS` (6 VMAF JND) yields eight representations for each video segment, while HLS ladders and `OPTE` encoding always have twelve representations. On average, `PPTE` (6 VMAF JND) yields ten representations for each video segment.

Considering temporal activity in live-streaming applications is crucial for achieving optimal video quality, and storage efficiency. Since h represents the temporal activity and is shown to have the strongest influence on the bitrate and VMAF prediction models compared to the other video complexity features (*cf.* Fig. 10), the correlation of h with the cumulative bitrate of all representations encoded using `JTPS` for different VMAF JND values (*i.e.*, 2, 4, and 6) is analyzed as shown in Fig. 12a, 12b, and 12c, respectively. The average R^2 score of h with the cumulative bitrate is 0.65. This is because, video segments with high temporal activity and fast-paced motion tend to have more temporal changes between frames, resulting in more information to be stored or transmitted. As a result, higher bitrates and larger file sizes are needed to maintain video quality. Similarly, the correlation of h with $\mid BDR_V \mid$ of the videos encoded using `JTPS` is analyzed for different VMAF JND values (*i.e.*, 2, 4, and 6) as shown in Fig. 12d, 12e, and 12f, respectively. The average R^2 score of h with $\mid BDR_V \mid$ is 0.51. In scenes with low temporal motion activity, where there is slower motion or minimal changes between frames, fewer bits are needed to represent the frames accurately. Hence, $\mid BDR_V \mid$ is high at low h values. However, $\mid BDR_V \mid$ is observed to be independent of the considered JND value. This is because the area under the RD curve using `JTPS` does not change based on JND values. To summarize, as h increases, *i.e.*, when there is an increase in temporal activity, the storage requirement increases. Furthermore, as h increases, the bitrate savings while maintaining the same VMAF decreases.

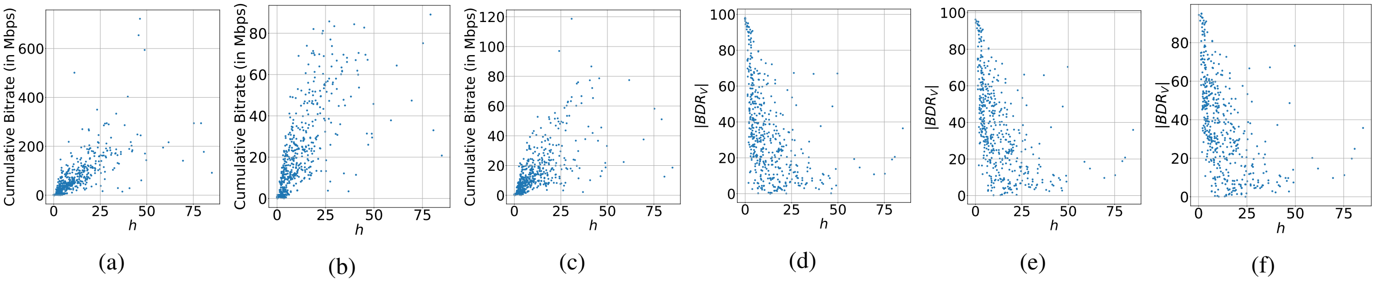
Finally, Table VI summarizes the bitrate saving results of the schemes in terms of BDR_P, BDR_V, and ΔS, the qualitative analysis results in terms of BD-PSNR and BD-VMAF, and encoding time saving (ΔT) compared to the HLS



(a) *Bunny_s000* (b) *Characters_s000* (c) *Eldorado_s000*

(d) *Eldorado_s005* (e) *HoneyBee_s000* (f) *Wood_s000*

Fig. 11: RD curves of representative video sequences (segments) (a) *Bunny_s000* (E =22.40, h=4.70, L=129.21), (b) *Characters_s000* (E =45.42, h=36.88, L=134.56), (c) *Eldorado_s000* (E=15.28, h=49.76, L=140.54) (d) *Eldorado_s005* (E =100.37, h=9.23, L=109.06), (e) *HoneyBee_s000* (E=42.93, h=7.91, L=103.00), (f) *Wood_s000* (E=124.72, h=47.03, L=119.57) using the HLS CBR encoding (green line), OPTE CBR encoding (brown line), PPTE encoding (olive line), HLS cVBR encoding (blue line), OPTE cVBR encoding (purple line), and JTPS encoding (red line). JND is considered as 6 VMAF in these plots.



(a) (b) (c) (d) (e) (f)

Fig. 12: Cumulative bitrate of all representations encoded using JTPS with (a) 2 VMAF JND, (b) 4 VMAF JND, and (c) 6 VMAF JND for various values of h , and $|BDR_V|$ results of JTPS encoding with (d) 2 VMAF JND, (e) 4 VMAF JND, and (f) 6 VMAF JND for various values of h, compared to HLS CBR encoding.

CBR encoding. Bruteforce encoding with JND of 2, 4, and 6 VMAF points is the best possible result when the predictions are 100% accurate. Hence, the corresponding results are the highest bound of the compression efficiency improvement (considering VMAF as the quality metric) compared to the HLS CBR encoding. The encoding time using the bruteforce method is 47 times higher than the HLS CBR encoding. OPTE CBR encoding yields bitrate savings of 17.28% and 22.79% to maintain the same PSNR and VMAF, respectively, compared to the HLS CBR encoding, along with a 0.07% cumulative increase in storage space required and a 9.74% cumulative increase in encoding time for various representations. This scheme yields the highest bitrate saving to maintain the same VMAF compared to the other CBR encoding schemes. PPTE scheme is analyzed for the JND values of 2, 4, and 6 VMAF points. With a target JND of 2 VMAF points, PPTE yields bitrate savings of 11.06% and 16.65% to maintain the same PSNR and VMAF, respectively, compared to the HLS CBR encoding, along with a 10.18% cumulative increase in storage space required and a 105.73% cumulative increase in encoding

TABLE VI: Average results of the encoding schemes compared to the HLS CBR encoding.

Method	BDR_P	BDR_V	BD-PSNR	BD-VMAF	ΔS	ΔT
Bruteforce (2 VMAF JND) [3], [13]	-23.09%	-43.23%	1.34 dB	10.61	-25.99%	4732.33%
Bruteforce (4 VMAF JND) [3], [13]	-28.15%	-42.75%	1.70 dB	10.08	-59.07%	4732.33%
Bruteforce (6 VMAF JND) [3], [13]	-25.36%	-40.73%	1.67 dB	9.19	-70.50%	4732.33%
OPTE CBR [11]	-17.28%	-22.79%	0.98 dB	3.79	0.07%	15.74%
PPTE (2 VMAF JND) [7]	-11.06%	-16.65%	0.87 dB	2.18	10.18%	105.73%
PPTE (4 VMAF JND) [7]	-10.44%	-15.13%	0.91 dB	2.39	-27.03%	10.19%
PPTE (6 VMAF JND) [7]	-12.94%	-17.94%	0.94 dB	2.32	-42.48%	-25.35%
HLS cVBR	-35.25%	-32.33%	2.09 dB	6.59	-9.39%	1.64%
OPTE cVBR	-34.42%	-42.67%	2.90 dB	9.51	-1.34%	62.73%
JTPS (2 VMAF JND)	-14.25%	-29.14%	1.36 dB	7.82	23.57%	184.62%
JTPS (4 VMAF JND)	-18.41%	-32.48%	1.41 dB	8.31	-56.38%	26.14%
JTPS (6 VMAF JND)	-18.80%	-32.59%	1.34 dB	8.34	-68.96%	-18.58%

time for various representations. The increase in storage space and encoding time is owed to the increase in the number of representations in the bitrate ladder when the JND value decreases. With a target JND of 4 and 6 VMAF points, the decrease in storage space requirement is observed as 27.03% and 42.48%, respectively. The overall encoding time increases by 10.19% for a target JND of 4 VMAF points, while it decreases by 25.35% for a target JND of 6 VMAF points.

It is observed that HLS cVBR encoding yields bitrate savings of 35.25% and 32.33% to maintain the same PSNR and VMAF, respectively, compared to the HLS CBR encoding, along with a 9.39% cumulative decrease in storage space and 1.64% cumulative increase in encoding time required for various representations. This result demonstrates that the compression efficiency of cVBR encoding is better than CBR encoding. Using OPTE cVBR encoding, bitrate savings of 34.42% and 42.67% to maintain the same PSNR and VMAF, respectively, are observed, compared to the HLS CBR encoding along with a 1.34% cumulative decrease in storage space requirement and a 62.73% cumulative increase in encoding time requirement. This scheme yields the highest bitrate saving to maintain the same VMAF compared to the other considered schemes. However, as observed in the RD figures, many representations are perceptually redundant, which wastes storage space. JTPS is observed to overcome this problem. With a target JND of 2 VMAF points, JTPS yields bitrate savings of 14.25% and 29.14% to maintain the same PSNR and VMAF, respectively, compared to the HLS CBR encoding, along with a 23.57% cumulative increase in storage space and a 184.62% cumulative increase in encoding time required for various representations. Similar to the observation for PPTE, when the JND value decreases, the number of representations in the bitrate ladder increases, causing an increase in storage space required. However, with a target JND of 4 and 6 VMAF points, the decrease in storage space requirement is observed as 56.38% and 68.96%, respectively. The overall encoding time increases by 26.14% for a target JND of 4 VMAF points, while it decreases by 18.58% for a target JND of 6 VMAF points.

V. Conclusions

This paper proposes a JND-aware two-pass cVBR per-title encoding scheme (JTPS) for adaptive live streaming applications. JTPS includes an optimized encoding bitrate ladder prediction algorithm, which estimates bitrate-resolution-CRF triples for a given video segment based on its spatial and temporal characteristics, using RF-based models. The bitrate ladder is predicted such that there is a perceptual difference of at least one JND between the representations in order to minimize the perceptual redundancy of the representations. Optimized CRF prediction for every representation in the bitrate ladder enables cVBR encoding. The experimental results show that, on average, JTPS yields bitrate savings of 18.80% and 32.59% to maintain the same PSNR and VMAF, respectively, compared to the CBR encoding of the reference HLS bitrate ladder with a negligible additional latency in streaming. This is accompanied by a cumulative decrease of 68.96% in storage space needed for various representations, and a cumulative decrease of 18.58% in encoding time, considering a JND of 6 VMAF.

In case the streaming service provider does not support per-title encoding schemes, the HLS cVBR encoding scheme can be used, where the bitrate-resolution pairs are fixed. Hence, the network architecture used for fixed bitrate-ladder encoding shall remain unaltered. If the streaming service provider supports dynamic resolution changes while maintaining a selected set of bitrates, OPTE cVBR encoding scheme is the best choice. Finally, if dynamic bitrate-resolution pairs are supported, JTPS offers the best storage reduction and improved compression efficiency.

In the future, JTPS can be extended to support Common Media Client Data (CMCD) [46], so that the encoding can be optimized based on the user profile, geolocation, subscription model, ratings, *etc.* In this way, context-awareness can be incorporated in JTPS.

References

[1] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 562–585, 2019.

[2] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[3] J. De Cock, Z. Li, M. Manohara, and A. Aaron, "Complexity-based consistent-quality encoding in the cloud," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1484–1488.

[4] M. Bhat, J.-M. Thiesse, and P. Le Callet, "A Case Study of Machine Learning Classifiers for Real-Time Adaptive Resolution Prediction in Video Coding," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[5] D. Silhavy, C. Krauss, A. Chen, A.-T. Nguyen, C. Müller, S. Arbanowski, S. Steglich, and L. Bassbouss, "Machine Learning for Per-Title Encoding," *SMPTE Motion Imaging Journal*, vol. 131, no. 3, pp. 42–50, 2022.

[6] D. Yuan, T. Zhao, Y. Xu, H. Xue, and L. Lin, "Visual JND: A Perceptual Measurement in Video Coding," *IEEE Access*, vol. 7, pp. 29 014–29 022, 2019.

[7] V. V. Menon, H. Amirpour, M. Ghanbari, and C. Timmerer, "Perceptually-Aware Per-Title Encoding for Adaptive Video Streaming," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2022, pp. 1–6. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICME52920.2022.9859744

[8] Y.-B. Lin and X.-M. Zhang, "Recent developments in perceptual video coding," in *2013 International Conference on Wavelet Analysis and Pattern Recognition*, 2013, pp. 259–264.

[9] T. Huang, R.-X. Zhang, and L. Sun, "Deep Reinforced Bitrate Ladders for Adaptive Video Streaming," ser. NOSSDAV '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 66–73. [Online]. Available: https://doi.org/10.1145/3458306.3458873

[10] V. V. Menon, H. Amirpour, M. Ghanbari, and C. Timmerer, "ETPS: Efficient Two-Pass Encoding Scheme for Adaptive Live Streaming," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 1516–1520.

[11] V. V. Menon, H. Amirpour, M. Ghanbari, and C. Timmerer, "OPTE: Online Per-Title Encoding for Live Video Streaming," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1865–1869.

[12] H. Amirpour, V. V. Menon, S. Afzal, M. Ghanbari, and C. Timmerer, "VCD: Video Complexity Dataset," in *Proceedings of the 13th ACM Multimedia Systems Conference*, ser. MMSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 234–239. [Online]. Available: https://doi.org/10.1145/3524273.3532892

[13] H. Amirpour, M. Ghanbari, and C. Timmerer, "DeepStream: Video Streaming Enhancements using Compressed Deep Neural Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[14] A. V. Katsenou, J. Sole, and D. R. Bull, "Content-gnostic Bitrate Ladder Prediction for Adaptive Video Streaming," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.

[15] A. Zabrovskiy, P. Agrawal, C. Timmerer, and R. Prodan, "FAUST: Fast Per-Scene Encoding Using Entropy-Based Scene Detection and Machine Learning," in *2021 30th Conference of Open Innovations Association FRUCT*, 2021, pp. 292–302.

[16] G. J. Sullivan, J. R. Ohm, W. J. Han, and 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.

[17] M. Shen, P. Xue, and W. Ci, "Down-sampling based video coding using super-resolution technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 755 – 765, Jul. 2011.

[18] Y. Reznik, K. Lillevold, A. Jagannath, and N. Barman, "Towards Efficient Multi-Codec Streaming," in *SMPTE 2022 Media Technology Summit*, 2022, pp. 1–16.

[19] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo, "The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment," in *2010 IEEE International Conference on Communications*, 2010, pp. 1–5.

[20] D. C. Knill and A. Pouget, "The bayesian brain: the role of uncertainty in neural coding and computation," *Trends in Neurosciences*, vol. 27, no. 12, pp. 712–719, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166223604003352

[21] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1385–1422, 1993.

[22] X. Zhang, W. Lin, and P. Xue, "A new DCT-based just-noticeable distortion estimator for images," in *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, vol. 1, 2003, pp. 287–291 Vol.1.

[23] Z. Chen and H. Liu, "Jnd modeling: Approaches and applications," in *2014 19th International Conference on Digital Signal Processing*, 2014, pp. 827–830.

[24] W. Lin and C.-C. Jay Kuo, "Perceptual visual quality metrics: A survey," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 297–312, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320311000204

[25] J. Zhu, S. Ling, Y. Baveye, and P. Le Callet, "A Framework to Map VMAF with the Probability of Just Noticeable Difference between Video Encoding Recipes," 2022. [Online]. Available: https://arxiv.org/pdf/2205.07565.pdf

[26] A. Kah, C. Friedrich, T. Rusert, C. Burgmair, W. Ruppel, and M. Narroschke, "Fundamental relationships between subjective quality, user acceptance, and the VMAF metric for a quality-based bit-rate ladder design for over-the-top video streaming services," in *Applications of Digital Image Processing XLIV*, vol. 11842, International Society for Optics and Photonics. SPIE, 2021, p. 118420Z. [Online]. Available: https://doi.org/10.1117/12.2593952

[27] C. Que, G. Chen, and J. Liu, "An Efficient Two-Pass VBR Encoding Algorithm for H.264," in *2006 International Conference on Communications, Circuits and Systems*, vol. 1, 2006, pp. 118–122.

[28] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[29] I. Zupancic, E. Izquierdo, M. Naccari, and M. Mrak, "Two-pass rate control for UHDTV delivery with HEVC," in *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1–5.

[30] S. Wang, A. Rehman, K. Zeng, J. Wang, and Z. Wang, "SSIM-Motivated Two-Pass VBR Coding for HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2189–2203, 2017.

[31] J. You and J. Korhonen, "Deep Neural Networks for No-Reference Video Quality Assessment," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 2349–2353.

[32] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 802–810.

[33] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3D LSTM: A Model for Video Prediction and Beyond," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1lKS2AqtX

[34] ITU-T, "P.910 : Subjective video quality assessment methods for multimedia applications," Nov. 2021. [Online]. Available: https://www.itu.int/rec/T-REC-P.910-202111-I/en

[35] V. V. Menon, C. Feldmann, H. Amirpour, M. Ghanbari, and C. Timmerer, "VCA: Video Complexity Analyzer," in *Proceedings of the 13th ACM Multimedia Systems Conference*, ser. MMSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 259–264. [Online]. Available: https://doi.org/10.1145/3524273.3532896

[36] M. King, Z. Tauber, and Z.-N. Li, "A New Energy Function for Segmentation and Compression," in *2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 1647–1650.

[37] Q. Cai, Z. Chen, D. O. Wu, and B. Huang, "Real-Time Constant Objective Quality Video Coding Strategy in High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2215–2228, 2020.

[38] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis (4th ed.)*. Wiley & Sons, 2006.

[39] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[40] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[41] Y. Chen, D. Murherjee, J. Han, A. Grange, Y. Xu, Z. Liu, S. Parker, C. Chen, H. Su, U. Joshi, C.-H. Chiang, Y. Wang, P. Wilkins, J. Bankoski, L. Trudeau, N. Egge, J.-M. Valin, T. Davies, S. Midtskogen, A. Norkin, and P. de Rivaz, "An Overview of Core Coding Tools in the AV1 Video Codec," in *2018 Picture Coding Symposium (PCS)*, 2018, pp. 41–45.

[42] P. K. Tiwari, V. V. Menon, J. Murugan, J. Chandrasekaran, G. S. Akisetty, P. Ramachandran, S. K. Venkata, C. A. Bird, and K. Cone, "Accelerating x265 with Intel® Advanced Vector Extensions 512," *White Paper on the Intel Developers Page*, 2018. [Online]. Available: https://www.intel.com/content/dam/develop/external/us/en/documents/mcw-intel-x265-avx512.pdf

[43] M. Bhat, J.-M. Thiesse, and P. L. Callet, "Combining Video Quality Metrics To Select Perceptually Accurate Resolution In A Wide Quality Range: A Case Study," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 2164–2168.

[44] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Sys-*

tems 30, 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[45] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, 2001.

[46] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, and R. Zimmermann, "Common Media Client Data (CMCD): Initial Findings," ser. NOSSDAV '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 25–33. [Online]. Available: https://doi.org/10.1145/3458306.3461444

Vignesh V Menon is a Ph.D. candidate at the Institute of Information Technology (ITEC), Alpen-Adria-Universität Klagenfurt (AAU), currently working on the ATHENA project. He received a B.Tech. degree in Electronics and Communication Engineering from Amrita Vishwa Vidyapeetham University, India, and an M.Sc. degree in Information and Network Engineering from KTH Royal Institute of Technology, Sweden, in 2016 and 2020, respectively. He worked as Software Engineer developing video encoding software solutions in MulticoreWare Inc., India, between 2016-2018 and in Divideon, Sweden, between 2018-2020. His research interests are video streaming, image, and video compression. Further information at https://vigneshvijay94.com.

Prajit T Rajendran is a Ph.D. candidate at Université Paris-Saclay, working on his doctoral research project in collaboration with CEA LIST and LNE France. He received his B.Engg degree in Electronics and Communication Engineering from Ramaiah Institute of Technology, Bangalore, India, in 2018 and subsequently an M.Sc. degree in Information and Network Engineering from KTH Royal Institute of Technology, Sweden, in 2020. His research interests include computer vision, deep learning, active learning, and human-in-the-loop artificial intelligence.

Christian Feldmann is a video coding engineer at Bitmovin working on next-generation video coding technologies in the most recent video coding standards, such as HEVC and AV1. After he studied computer engineering at RWTH University Aachen, he completed his doctoral degree (Ph.D.) at the Institut für Nachrichtentechnik (Institute for Communication Technologies) in Aachen. With his detailed experience in video coding, he is developing video coding technologies for the future of video coding. Christian participates in the standardization activities of the Alliance for Open Media (AOMedia) and the Moving Picture Experts Group (MPEG).

Klaus Schoeffmann is an associate professor at the Institute of Information Technology (ITEC). He received his Ph.D. degree and Habilitation (venia docendi) from Klagenfurt University in both 2009 and 2015, respectively, in computer science. He is currently an Associate Professor with the Institute of Information Technology (ITEC), Klagenfurt University, Klagenfurt, Austria. His research focuses on video analytics and interactive multimedia systems, particularly in the medical domain. He has co-authored more than 110 publications on various topics in multimedia. He has co-organized several international conferences, workshops, and special sessions in the field of multimedia. Furthermore, he is a co-founder of the Video Browser Showdown (VBS), a member of the ACM, and a regular reviewer for international conferences and journals in multimedia.

Mohammad Ghanbari (M'78–SM'97–F'01, LF'14) is an Emeritus Professor at the School of Computer Science and Electronic Engineering, University of Essex, United Kingdom. He is currently involved in the ATHENA Project at the Alpen-Adria-Universität Klagenfurt (AAU), Austria. He is internationally best known for his pioneering work on layered video coding, which earned him IEEE Fellowship in 2001, and he was also promoted to IEEE Life Fellow in 2014. He has registered for thirteen international patents and published more than 770 technical papers on various aspects of video networking, many of which have had fundamental influences in this field. These include video/image compression, layered/scalable video coding, transcoding, motion estimation, and video quality metrics. He is the author and co-author of 8 books, and his book video coding: an introduction to standard codecs, published by IET press in 1999, received the Rayleigh prize as the best book of the year 2000 by IET. He was one of the founding Associate Editors of IEEE Trans on Multimedia from 1999 to 2002.

Christian Timmerer (M'08-SM'16) is a full professor of computer science at Alpen-Adria-Universität Klagenfurt (AAU), Institute of Information Technology (ITEC), and he is the director of the Christian Doppler (CD) Laboratory ATHENA (https://athena.itec.aau.at/). His research interests include multimedia systems, immersive multimedia communication, streaming, adaptation, and quality of experience, where he co-authored eight patents and more than 300 articles. He was the general chair of WIAMIS 2008, QoMEX 2013, MMSys 2016, and PV 2018 and has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, ALICANTE, SocialSensor, COST IC1003 QUALINET, ICoSOLE, and SPIRIT. He also participated in ISO/MPEG work for several years, notably in MPEG-21, MPEG-M, MPEG-V, and MPEG-DASH, where he served as standard editor. In 2012 he cofounded Bitmovin (http://www.bitmovin.com/) to provide professional services around MPEG-DASH, where he holds the position of the Chief Innovation Officer (CIO) — Head of Research and Standardization. Further information at http://timmerer.com.