NCOD: Near-Optimum Video Compression for Object Detection

Ardavan Elahi[1], Ali Falahati[2], Farhad Pakdaman[3], Mehdi Modarressi[1], Moncef Gabbouj[3]

[1]Department of Computer, School of Electrical and Computer Engineering, College of Eng., University of Tehran, Iran

[2]Avanco.tech

[3]Faculty of Information Technology and Communication Sciences, Tampere Univesrsity, Finland

***Abstract*—Nowadays, machine vision applications play an essential role in our everyday life. With the emergence of technologies like smart cities, Internet of things (IoT), and 5G, the amount of produced video data at the edges and remote nodes has exploded. Since for a considerable portion of the captured video the target is a machine learning task, rather than a human audience, transmission of videos in such applications requires efficient video compression tailored for machine vision. However, existing compression solutions are optimized for human vision. This paper presents a methodology to optimize an existing video compression standard, HEVC, for a machine vision task, Object Detection (OD). To this end, (1) a dataset of compressed videos, including several compression-ratios and their corresponding OD performance is collected to enable modeling, (2) A trade-off point (knee-point) between bitrate and OD performance is defined, that finds the point after which no major improvements will be achieved, (3) an extensive set of features were extracted and studied to model this point, via a practical machine learning method. The resulting solution can predict the knee-point with MAE=1.28, resulting in a ΔRecall of only 0.012 and bitrate reduction of 86.56%, compared to OD with very high-quality video.**

Keywords—Video coding, Video coding for machine (VCM), CRF, Object Detection,

I. INTRODUCTION

Computer vision and image/video processing play a vital role in modern intelligent world. Enabling technologies, such as smart cities, the Internet of Things (IoT), autonomous driving, and AR/VR, leverage the state-of-the-art computer vision algorithm to thrive. In most cases, the recording devices are placed at the edges and the compressed video is transmitted over the network for further processing - as sensor nodes often poses limited processing power, and/or analysis should be done centrally. Computing paradigms like edge-to-cloud [1] and collaborative intelligence [2] are among such cases which, first compress the videos by standard video codecs at the edge and then transmit them to a cloud server or other nodes. This paradigm is called Compress-then-Analyze[3].

With the advent of emerging technologies, most videos will be consumed by machine vision, which processes videos different way than human vision. However, algorithms used in existing video standards are based on Human Visual System (HVS). These computer vision systems usually require efficient power and energy consumption and operate in bandwidth-limited environments. This led to the emergence of the Video Coding for Machines ad-hoc group in 2019 in the MPEG standardization group [4]. The quest is to develop and standardize new video compression standards, suitable for machine vision. Since the introduction of VCM, new architectures and algorithms have been proposed. Yang et al. [5] categorized the proposed solutions in two broad categories: improving existing codecs to comply with VCM objectives, or developing end-to-end learned codecs for VCM usage. Developing video compression algorithms for Object Detection (OD) is among the efforts in the first category. In [7] and [8] saliency map-based video coding schemes were presented to efficiently compress videos while preserving object detector performance. In [9], a novel bit allocation strategy for HEVC was proposed to adaptively allocate bits based on saliency for yolov3 [10]. Authors in [11] present an algorithm to adjust the quantization for blocks of VVC based on the contents of each block, to efficiently compress videos for OD. While these methods achieve some improvements, (1) they need to adjust or modify codecs, and (2) they do not consider the bitrate-performance trade-off in existing codecs, which leads to sub-optimum compression. Moreover, developing new standards/codecs (1) take considerable time for practical deployment (2) may require spending extra costs as royalty fees, and (3) requires developing hardware/software to comply with the new standard.

With this motivation, this paper aims to provide an efficient video coding solution for machine vision, which is based on existing standards and can be deployed with minimum effort and cost. A methodology is proposed to set the rate-distortion parameter (CRF) of an existing codec to a Near-optimal Compression point, for Object Detection (NCOD). As video compression is quite content-dependent, it affects OD on different videos differently. To solve this problem, (1) a dataset is collected that includes several video scenes compressed with different HEVC configurations, and their corresponding OD performance using yolov4 [6], (2) bitrate-performance curves have been studied and the "knee-points" have been identified and formulated as an intuitive trade-off point between rate and OD performance, (3) extensive features have been extracted and their correlation to the task has been investigated. Feature selection and reduction techniques were used to find the best feature set, (4) finally, a machine learning model has been trained to accurately predict the knee-point for each video. The contributions of the paper are summarized as follows:

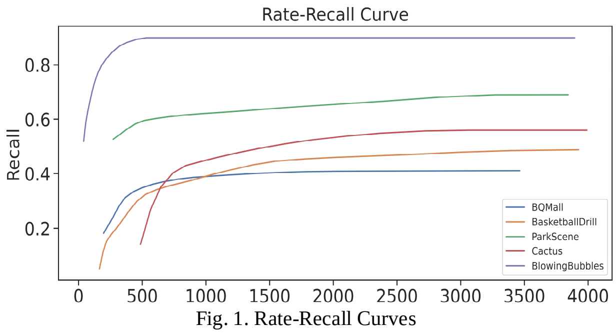

Fig. 1. Rate-Recall Curves

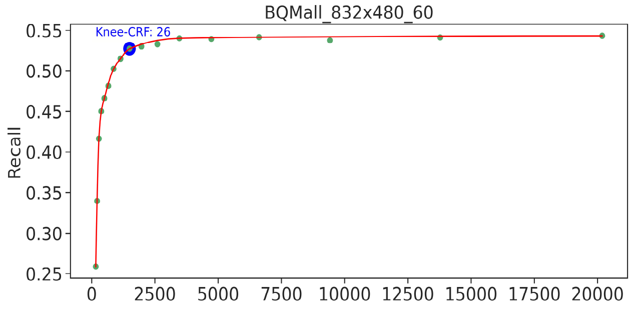

Fig. 2. Convex Hull of Rate-Recall for BQMall Sequence

- A dataset of compressed video scenes, their corresponding OD performance, and various representative features, is developed that facilitates studying compression for the OD task. The dataset consists of an overall of 4148 data points and will be released on the project webpage: https://github.com/researchVCM/NCOD.
- The knee-point definition is suggested for OD, that defines the compression point above which no considerable gain of performance can be achieved.
- The choice of features for modeling the knee-point is studied, and feature extraction, selection, and reduction techniques are used to find the best set of features.
- The knee-point prediction is solved as a regression task. The final solution is able to configure HEVC codec to a near-optimum point, just receiving the input video, with no modifications required for the encoder or decoder.

The rest of this paper is organized as follows. Section II details the proposed methodology. Section III presents the experimental results. Finally, Section IV concludes the paper.

II. Methodology

A. Overview

The main goal of this work is to model the best trade-off point between bitrate and OD performance, using intrinsic features of raw video sequences. To do so, we select the CRF (which loosely corresponds to the Quantization Parameter value - widely used to control bitrate in video compression) as the main knob to configure the compression intensity. Libx265 [12] is used as a widely used open-source HEVC codec. In many cases, object detection models fail to detect several objects in a video and compression artifacts tend to intensify this [1]. In such a condition, precision is not the best metric to measure the performance, and recall becomes a more intuitive metric. Hence, we select recall as the main accuracy metric, for modeling. To find a good trade-off between bitrate and performance, we define the knee-point, as the compression point after which no major gain of performance can be achieved. We demonstrate that this point is highly content-dependent. Hence, to build a model that can predict this point for each video, first, a dataset is collected through exhaustive experiments on SFU-HW-Objects-v1[13] videos. To build a more thorough dataset and enable a robust learning, the dataset is augmented to include modified versions of each video. Then, the knee-points are identified for each case, and the final dataset is used for model training. An extensive set of features corresponding to various video characteristics are extracted. Then, feature selection and reduction techniques are used to reduce the feature size to comply with the data size. Finally, the knee-points are modeled and predicted in a regression task. The following steps, detail different parts of this methodology.

B. Rate-Performance Trade-off and Knee-points

Fig. 1 depicts the achieved recall of yolov4 [6] across different bitrates (different CRFs), for some videos in SFU-HW-Objects-v1 dataset. Bitrate-Recall curves depict how decreasing the bitrate reduces OD performance. It can be observed that for each video, there is a certain CRF point after which recall starts to decline rapidly. This point, named Knee-point, differs for each video and is highly content-dependent.

To predict the knee-point of a curve, these curves require to be Pareto Efficient [14][15]. To make this applicable, we compute the convex hull of the Bitrate-Recall curves, as in [15]. Then we apply piecewise cubic Hermite interpolation

TABLE I. Details of Selected Video Sequences

Video Name	Resolution	Frame Count	Frame Rate
Traffic	2560x1600	150	30
PeopleOnStreet	2560x1600	150	30
BasketballDrive	1920x1080	500	50
Cactus	1920x1080	500	50
ParkScene	1920x1080	240	24
BQMall	832x480	600	60
BasketballDrill	832x480	500	50
PartyScene	832x480	500	50
BQSquare	416x240	600	60
BasketballPass	416x240	500	50
BlowingBubbles	416x240	500	50
KristenAndSara	1280x720	600	60
Johnny	1280x720	600	60

TABLE II. Selected Video Features

Source	Features
VCA[20]	avgU, energyU, avgV, energyV, Spatial complexity (E), Temporal complexity (h), Brightness (L)
AGH[21]	Interlace, Noise, Blockloss, Spatial Activity, Blur, Flickering, Contrast, TemporalAct, Blockiness, Exposure
SITI[22]	Spatial information (SI), Temporal information (TI)
CAMBI[23]	Cambi
Quat[24]	Noise, Blur, Blockiness, Colorfulness, contrast, cubrow.[0, 0.3, 0.5, 0.6, 1.0], cubcol.[0, 0.3, 0.5, 0.6, 1.0], FFT, Movement, Saturation, Staticness, Temporal, Tone, Similarity to half resolution

[19] to estimate the interim points of the curve. The convex hull of a set of points results in a subset of the points which wraps a band around all the points. Fig. 2 depicts the convex-hull for BQMall. Each green dot corresponds to bitrate and recall at a specific CRF value. The orange line shows the convex hull of the encoding points.

Next, the knee-point of the convex hull should be computed. The term “Knee-point” in cost-benefit analysis refers to a point at which enhancing some adjustable parameters no longer yields significant performance improvement [16]. Authors in [16] provide the mathematical definition of this point, and present an algorithm called “Kneedle” to find the point in an application-independent manner. In Fig. 2 the blue dot with the value of 26 is the knee-point. It suggests that encoded videos with CRF values below 26 (all points right of the knee-point) do not improve recall in a cost-efficient way. Intuitively speaking, a CRF value of 26 is “good enough” for this video to achieve a highly compressed video and yet near-optimum OD performance. The difference between the recall of CRF=26 and CRF=10 (representing very high-quality video) is only 0.012 while bitrate reduction is 92.48%. These numbers for the average of all data are 0.027 and 89.82%, respectively.

C. Dataset Collection

To collect a dataset of compressed videos and their corresponding OD performance, exhaustive encodings across multiple CRF values were performed to achieve low-compressed to highly compressed samples. Thirteen video sequences of SFU-HW-Objects-v1 (the main dataset recommended in VCM development, containing raw video quality) were used for this. We skipped five other videos in this dataset due to very poor yolov4 performances. Table I lists the details of these videos.

Developing an accurate machine-learning model requires more than thirteen data points. To solve this issue, we largened our dataset with two data augmentation techniques. First, all of these thirteen sequences were segmented into 100-frames-length videos, leading to 60 videos. Next, to mimic the intrinsic noises of a camera, we blurred these frames with three different blurring filters using OpenCV v4.6.0 [17] and FFMPEG [18]. The BilateralFilter from OpenCV is used with filter parameters set to (11, 21, 7), (11, 41, 21), and (11, 61, 39), where the numbers in sets represent diameter, sigma color, and sigma space of the filter, respectively. After these steps, we achieve 244 videos, which we call video chunks.

Each video chunk is compressed and decompressed with CRF values between 10 to 42 with a step size of 2. This results in 17 videos for each of the 244 video chunks, i.e., 4148 videos. Next, each of the decoded videos were processed with the yolov4 model with the non-maximum suppression parameters of Confidence=0.6 and Threshold=0.8; and IoU Threshold=0.7. Relevant metrics such as bitrate, OD recall and precision, as well as encoding and decoding times were collected for all video chunks, and convex hull and knee-points were obtained as described in II. B. Fig. 3 shows the distribution of the knee-points for all 244 video chunks. Due to the lack of knee-points outside the range of (23, 30), we select only videos within this range for the train and the test stages. This shows that the existing datasets in academia for research on VCM are not diverse in spatial and temporal characteristics.

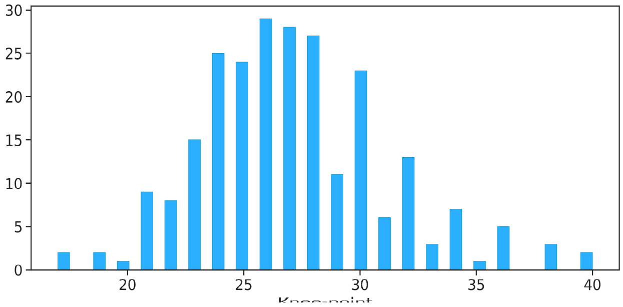

Fig. 3. Distribution of Knee-points

D. Machine Learning Flow

Feature Extraction: In order to predict the CRF value corresponding to the knee-point in the Rate-Recall curves, we first need to extract Spatio-temporal features of raw videos. We explored 42 features from 5 different sources widely used in the literature, including Video Complexity Analyzer (VCA) [20], video quality indicators (AGH) [21], SITI [22], CAMBI [23], and the Quat Analysis Tools (QUAT) [24]. Table II lists all these features. Due to the resolution dependency of these features, uncompressed videos were scaled to 1920x1080 resolution prior to feature extraction. Each feature estimates one value per frame, therefore we use temporal pooling to extract the features for each video. For each feature, we compute mean, standard deviation, skewness, kurtosis, interquartile range, 0.25 quantile, 0.75 quantile, the first and last value, and the minimum and maximum values over all frames of a video sequence. This leads to 11 pooled values per feature resulting in 462 values per video. Fig. 4 shows the distribution of SI and TI as two examples, demonstrating that the dataset covers a wide variety of video content.

Feature Selection and Reduction: To prevent overfitting, we select 100 features with the highest Pearson correlation coefficient value with the dependent variable (CRF value of the knee point). After that, the multicollinearity among these features is removed and using the Principal Component Analysis (PCA), the 30 components with the highest amount of information are selected for model training.

Model Training: After evaluating multiple classifications and regression methods, we observe that regression performs better for predicting the knee-points. We selected the AdaBoost regressor [25] as the best method with the lowest mean absolute error (MAE). The AdaBoost is a meta-algorithm that boosts the performance of base estimators, in our case decision-tree, by ensemble learning. Video chunks corresponding to 10 out of 13 original video scenes (and all corresponding processed versions, total 128 video chunks) were used for training, and three original video scenes (BasketballPass, BasketballDrill, and BlowingBubbles) were kept only for final evaluations. We trained a model using 10-fold cross-validation, making sure each video chunk and its blurred versions appear in the same fold for fair results during the cross-validation. We also used chunks from a video sequence only in one of the test or train set to avoid leaking information from the test set to the train set. For training, 50 estimators are ensembled, using a linear combination of losses for boosting [26], and a learning rate of 1.0.

III. Experimental Results

As mentioned before, 54 video chunks corresponding to three video scenes are considered as test video sequences. The evaluations are performed as follows.

Model Accuracy: We tested the model on the test video chunks. The prediction error for knee-points can be observed in Fig. 5. For the test set, MAE is 1.28, and RMSE is 1.66 on average, which indicates a very high prediction accuracy. To further assess the quality of the predicted points we measured the difference between the bitrate and recall of the predicted knee-points, and those of the ground truth knee-points, namely ΔRate and ΔRecall respectively. On average, ΔRate and ΔRecall are only 0.73% and 0.132, respectively. Please note that an ideal predictor results in zero delta values.

Method Performance: Bitrate, recall, precision, decoding time, and encoding time are selected as the comparison metrics. As the proposed methodology predicts the optimal CRF points, the ground truth knee-point is set as the baseline. Also, the following comparisons with existing solutions are considered: (1) Just Noticeable Difference (JND) [27] aims to find the compression point after which distortions become noticeable; however for human and not machines. The JND prediction method presented in [27] is selected for comparing the OD performance, (2) Most computer vision systems perform OD on raw videos or very high bitrate compressed videos, to get high accuracy. Hence, we compare the bitrate required for OD task, with the high-quality case of CRF=10. This point corresponds to a very high-quality image, after which no fluctuations on the bitrate-recall curves are observed. Hence, it could be considered similar to uncompressed video performance, but with lower bitrate. As already mentioned in section II, the OD performance of the knee-point is only 0.027 smaller than those of CRF=10.

Table III compares NCOD and JND, against the ground truth knee-point, for ΔRecall and ΔPrecision. As the knee-point is considered as the optimum point of compression for OD, ΔRecalls and ΔPrecisions closer to zero are better. It can be observed that NCOD achieves on average 0.012 ΔRecall and 0.007 ΔPrecision, which are the smallest among the two. It is important to note that (1) the model has been trained on recall and not precision; hence, a small ΔPrecision confirms the generalization of the model, and (2) as mentioned earlier, precision changes unpredictably with CRF. Hence, ΔPrecision does not fully reflect the performance.

Table IV compares NCOD and JND against CRF10 w.r.t bitrate reduction, encoding time, and decoding time. It can be observed that NCOD achieves 86.56% bitrate reduction with the baseline of CRF=10. Moreover, as encoding and decoding are less complex in lower bitrates, encoding and decoding of NCOD are 46.14% and 54.88% faster. JND gains similar results with NCOD, however, with lower OD performance.

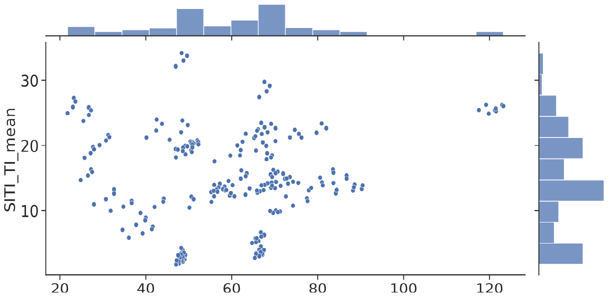

Fig. 4. Temporal and Spatial Distribution of the Dataset

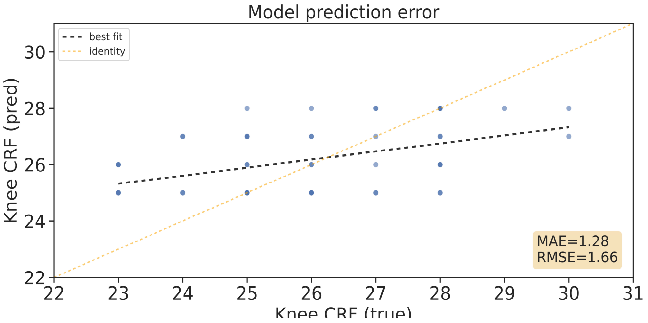

Fig. 5. Prediction Error of the Model

TABLE III. Results of NCOD and JND against Knee-point

Video Sequence	**ΔRecall**		**ΔPrecision**	
	NCOD	***JND***	***NCOD***	***JND***
BasketballDrill	0.008	0.031	0.010	0.014
BasketballPass	0.016	0.020	0.008	0.010
BlowingBubbles	0.015	0.023	0.007	0.005
Average(all chunks)	0.012	0.223	0.007	0.011

TABLE IV. Results of NCOD and JND against CRF-10

Video Sequence	**Birate Reduction%**		**EncTime Reduction%**		**DecTime Reduction%**	
	NCOD	***JND***	***NCOD***	***JND***	***NCOD***	***JND***
BasketballDrill	87.73	91.81	49.61	55.33	57.56	62.71
BasketballPass	87.01	86.29	42.62	41.89	55.80	54.93
BlowingBubbles	87.92	90.44	55.5	59.04	63.87	67.04
Average(all chunks)	86.56	88.48	46.14	48.4	54.88	57.08

IV. Conclusion

This paper presented a methodology to optimize video compression for a machine vision task, namely object detection. A trade-off point (knee-point) between bitrate and OD performance has been defined, that finds the point after which no major improvements will be achieved. A dataset was collected that enables learning a model, to predict the knee-points. Finally, the problem was solved as a regression task, where several features were used to model the knee-point. It was observed that (1) recall is a better choice for modeling the performance, as compression artifacts affect the precision unpredictably, (2) the knee-point achieve a recall very close to the uncompressed, while requiring a much lower bitrate, (3) knee-point is content dependent and hence, several video features have been used for accurate modeling. The final method can predict the knee-point with an average MAE of only 1.28, which leads to a ΔRecall of only 0.012, while reducing the bitrate by 86.56% compared to baseline methods that use a CRF-10.

Extending the collected dataset with (1) more video samples and (2) covering more diverse spatial and temporal complexities is considered as an effective future step. Moreover, we believe similar methodology can be applied for other machine vision tasks, such as object tracking and segmentation. More investigations are required to study different aspects of such extension.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No [101022466].

References

[1] W. Gao et al., "Digital Retina: A way to make the city brain more efficient by visual coding," IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 11, pp. 4147–4161, Nov. 2021.

[2] I. V. Bajić, W. Lin, and Y. Tian, "Collaborative intelligence: challenges and opportunities," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Jun. 2021, pp. 8493–8497.

[3] S. Wang, Z. Wang, S. Wang, and Y. Ye, "End-to-end compression towards machine vision: network architecture design and optimization," IEEE Open Journal of Circuits and Systems, vol. 2, pp. 675–685, 2021.

[4] L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: a paradigm of collaborative compression and intelligent analytics," IEEE Trans. Image Process., vol. PP, Aug. 2020.

[5] W. Yang, H. Huang, Y. Hu, L.-Y. Duan, and J. Liu, "Video coding for machine: compact visual representation compression for intelligent collaborative analytics," arXiv preprint arXiv:2110.09241, 2021.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[7] L. Galteri, M. Bertini, L. Seidenari, and A. Del Bimbo, "Video compression for object detection algorithms," in 2018 24th International Conference on Pattern Recognition (ICPR), Aug. 2018, pp. 3007–3012.

[8] K. Fischer, F. Fleckenstein, C. Herglotz, and A. Kaup, "Saliency-driven versatile video coding for neural object detection," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Jun. 2021, pp. 1505–1509.

[9] Q. Cai, Z. Chen, D. O. Wu, S. Liu, and X. Li, "A novel video coding strategy in HEVC for object detection," IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 12, pp. 4924–4937, Dec. 2021.

[10] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[11] M.-J. Kim and Y.-L. Lee, "Object detection-based video compression," Applied Sciences, vol. 12, no. 9, p. 4525, Apr. 2022.

[12] (X265) https://www.videolan.org/developers/x265.html. Accessed 05 November 2022.

[13] H. Choi, E. Hosseini, S. Ranjbar Alvar, R. A. Cohen, and I. V. Bajić, "A dataset of labelled objects on raw video sequences," Data Brief, vol. 34, p. 106701, Feb. 2021.

[14] (Pareto Efficiency) https://en.wikipedia.org/wiki/Pareto_efficiency, Accessed 05 November 2022.

[15] (Netflix PerTitle Optimization) https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2 , Accessed 05 November 2022

[16] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a 'kneedle' in a haystack: detecting knee points in system behavior," in 2011 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, USA, Jun. 2011, pp. 166–171.

[17] (OpenCV) www.opencv.org , Accessed 05 November 2022.

[18] (FFMPEG) www.ffmpeg.org , Accessed 05 November 2022.

[19] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," SIAM J. Numer. Anal., vol. 17, no. 2, pp. 238–246, Apr. 1980.

[20] 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, Athlone, Ireland, Aug. 2022, pp. 259–264.

[21] J. Nawała, L. Janowski, and M. Leszczuk, "Modeling of quality of experience in no-reference model," J. Telecommun. Inf. Technol., vol. 2, no. 2017, pp. 11–17, Jul. 2017.

[22] ITU-T Recommendation, P910, "Subjective video quality assessment methods for multimedia applications", International telecommunication union, 1999.

[23] P. Tandon, M. Afonso, J. Sole, and L. Krasula, "CAMBI: contrast-aware multiscale banding index," in 2021 Picture Coding Symposium (PCS), Jun. 2021, pp. 1–5.

[24] S. Göring, R. R. R. Rao, B. Feiten, and A. Raake, "Modular framework and instances of pixel-based video quality models for UHD-1/4K," IEEE Access, vol. 9, pp. 31842–31864, 2021.

[25] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. System Sci., vol. 55, no. 1, pp. 119–139, Aug. 1997.

[26] H. Drucker, "Improving regressors using boosting techniques". In International Conference on Machine Learning (ICML), pp. 107-115, 1997.

[27] S. Nami, F. Pakdaman, and M. R. Hashemi, "Juniper: A jnd-based perceptual video coding framework to jointly utilize saliency and JND," in 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Jul. 2020, pp. 1–6.