PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Machine-learning-based tuning of encoding parameters for UGC video coding optimizations

Chaudhari, Gaurang, Koba, Igor, Katsavounidis, Ioannis, Reddy, Harikrishna

Gaurang Chaudhari, Igor Koba, Ioannis Katsavounidis, Harikrishna Reddy, "Machine-learning-based tuning of encoding parameters for UGC video coding optimizations," Proc. SPIE 11842, Applications of Digital Image Processing XLIV, 1184208 (1 August 2021); doi: 10.1117/12.2595404

SPIE.

Event: SPIE Optical Engineering + Applications, 2021, San Diego, California, United States

Machine learning based tuning of encoding parameters for UGC video coding optimizations

Gaurang Chaudhari, Igor Koba, Ioannis Katsavounidis, Harikrishna Reddy
Facebook Inc, 1 Hacker Way, Menlo Park, CA, USA 94025

ABSTRACT

In the era of COVID-19 pandemic, videos are very important to the billions of people staying and working at home. Two-pass video encoding allows for a refinement of parameters based on statistics obtained from the first pass. Given the variety of characteristics in user-generated content, there is opportunity to make this refinement optimal for this type of content. We show how we can replace the traditional models used for rate control in video coding with better prediction models with linear and nonlinear model functions. Moreover, we can utilize these first-pass statistics to further refine the traditional encoding recipes that are typically used for all input video sequences. Our work can provide much-needed bitrate savings for many different encoders, and we highlight it by testing on typical Facebook video content.

Keywords: video encoding, user generated content, UGC, quantization parameter, Qp, VP9, linear regression, rate control, first pass statistics, coding efficiency

I. INTRODUCTION

Many emerging works in video coding have applied machine learning-based algorithms to further improve the coding efficiency or quality, to reduce complexity or make faster decisions in reducing the search space for optimal encoding. In this paper, we want to introduce the application of machine learning in the rate control aspect of video encoding, particularly in selecting the proper quantization parameter (Qp).

Rate control (RC) is an essential component of video encoding. It can maximize video quality by optimally distributing the available bits to meet a bandwidth or file size constraint for the duration of the video stream. A typical rate control method relies on bit usage statistics (past frame bits), Qp (Quantization Parameter), target bits, buffer constraints and frame statistics (encoded quality, past and present frame encoding complexity, etc) as inputs and outputs the Qp for the next encoded video frame. Qp for a typical video encoder determines a tradeoff between residual error (distortion) and amount of encoded bits for the frame, with larger Qp values corresponding to lower quality/lower bitrates. It is important to note that, while Qp value for the next frame is decided by rate control conditioned on the prior statistics and encoder data, the final outcome of the decision, i.e., residual error (distortion) and number of coded bits of the frame is not known until encoder finishes encoding a given video frame. Only then does the frame encoder update information in the rate control unit, used for decision purposes for subsequent frames. In a typical video encoder, a rate control mechanism shown in Fig. 1 can be described in the following high-level sequence:

1. Estimate frame complexity and target bits for the frame.
2. Choose a Qp that gives the best trade-off between controlling rate and overall quality.
3. Encode the frame.
4. Update number of encoded bits and other frame statistics.

Applications of Digital Image Processing XLIV, edited by Andrew G. Tescher, Touradj Ebrahimi, Proc. of SPIE
Vol. 11842, 1184208 · © 2021 SPIE · CCC code: 0277-786X/21/$21 · doi: 10.1117/12.2595404

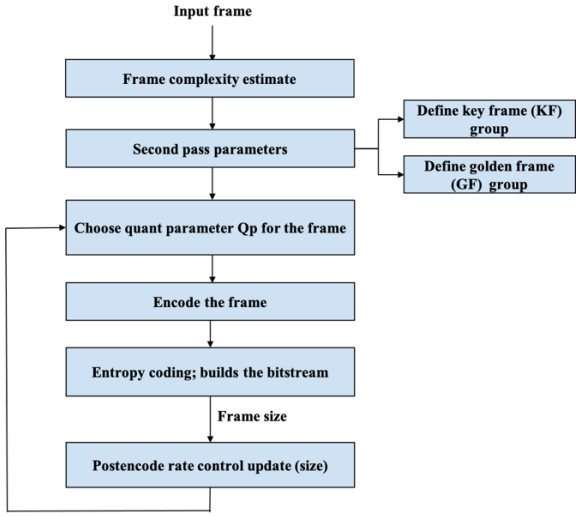

Figure 1. Algorithm of rate control in conventional VP9 encoder.

As rate control estimates the frame complexity, it also decides on the group of pictures (GOP) and the key aspects of GOP structure like finding the next Intra-coded frame, known as keyframe in VP9 standard, user-enforced GOP interval constraints, target bits for the frames in the group, quality boost (lower Qp value for certain frames than the average value of a set of frames) calculation to distribute available bits across the group, etc. Depending on use cases, first-pass analysis data for a number of frames is completed ahead of time and used by rate control algorithms.

While the method proposed in this paper may be potentially used for any type of video encoder, we will describe it with respect to the libvpx[1] implementation of VP9 standard[2][3] as introduced in Section II. Section III provides an overview of how we can use a machine-learning model based on first pass statistics to better predict the Qp for boosted frames, followed by section IV which presents the results of the proposed method.

II. LIBVPX FRAME BOOST

Libvpx is an open source VP9 encoder implementation frequently used as a reference by developers. This paper first describes a particular rate control implementation in the libvpx library followed by advancements done in the proposed implementation. Libvpx has a 2-pass encoding method, in which a very fast first-pass analysis of the entire video stream is done, and the resulting statistics are used for deciding the second pass parameters (computing target frame sizes and planning bit distribution across the stream). The loop in Fig. 1 is repeated for every encoded frame during the second encoding pass.

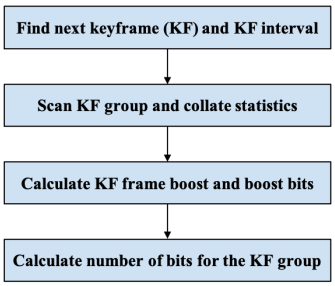

Figure 2. Defining key frame (KF) group

As rate control makes decisions on second pass parameters, it tries to define the key frame (KF) group as shown in Fig. 2. Once the next KF and the KF interval is decided based on scene-cut and user constraints, the number of bits to assign to the KF group is calculated based on the remaining bits and the relative complexity of the section. The KF group is scanned to collect and accumulate various statistics which help in determining the number of bits to spend on the KF group, determining the static sections, and deciding the KF boost. The frame boost is calculated as in (1).

frame_boost = (err_per_mb * active_area) / inter_error (1)

The KF boost calculations are based on empirical data for error per macroblock (err_per_mb) where the term active_area offers a discount for image masks, bars and other zero-energy areas. Moreover, the baseline numbers used in Qp correction and scaling are also empirically derived. Depending on the key frame boost, the boost bits are calculated, i.e the number of bits to allocate for the keyframe itself. Finally, the number of bits that should be assigned to the KF group are recalculated by adjusting the key frame bits.

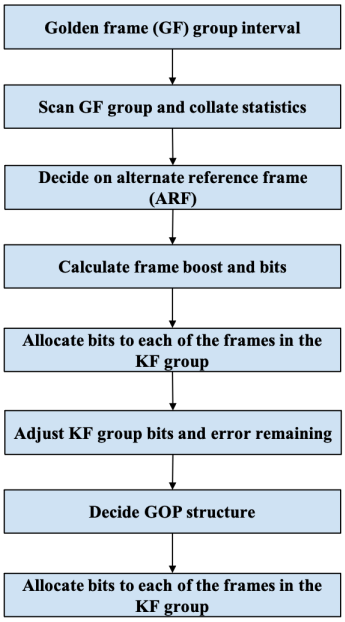

Figure 3. Definition of golden frame (GF) group

A KF group may contain multiple golden frame (GF) groups and for each GF group (golden frame is a frame from the arbitrarily distant past[4]), the process shown in Fig. 3 is followed. For a new GF group, the maximum and minimum intervals for the GF group are set and the GF group is scanned to collate statistics of frames in the GF group. These statistics help in determining static sections, deciding whether to use an alternate reference frame (ARF), and deciding the frame boost. Similar to KF, the boost calculations are based on empirical data for error per macroblock and Qp correction and scaling. Depending on the KF group bits, the GF group error and the remaining KF group error, the total number of bits are allocated to the whole GF group. Depending on the boosted frames, the extra bits to be used for boosted frames in the group are calculated and the number of bits to be assigned to the GF group are adjusted. Based on this, the KF group bits and remaining error are adjusted as well. The layer depth and order within the GF group is decided, and bits are allocated for each of the frames in the GF group.

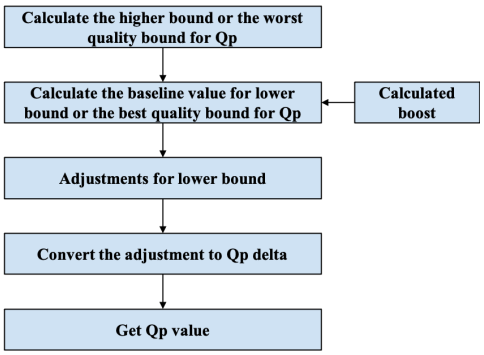

Figure 4. Quantization parameter (Qp) decision

The step "Choose quant parameter" in Fig. 1 calculates the quantization parameter for the next frame. It relies on the frame size prediction to estimate bits at a certain value of Qp, which in turn relies on predicted bits per macroblock. As shown in Fig. 4, maximum and minimum bounds are computed for a certain Qp. The higher bound or the worst quality bound is calculated during the golden frame group decision. Based on the recent history, the expectations of bits per macroblock are adjusted to pick a maximum value that will be high enough to encode the content at the given rate. The lower bound or the best quality bound is calculated on top of the worst quality and the calculated frame boost during the KF/GF group decisions. For KF, the adjustment factor for the lower bound is computed based on an empirical equation involving a motion measure. For GF, the lower bound is linearly fitted depending on the layer depth. The adjustment factor is then converted to Qp delta and converted to the final Qp value.

The step "Encode frame" in Fig. 1 does most of the actual frame encoding such as mode decision, transform coefficients, and residual calculation for each superblock in the frame. The subsequent "Entropy coding" step is a final step, where previously generated mode decision and transform coefficients are packed using VP9's entropy coding method according to the standard. VP9 uses a tree-based boolean non-adaptive binary arithmetic encoder to encode all syntax elements. It is important to note that "Encode frame" and "Entropy coding" are integral parts of the frame encode process. The final step is a post-encode update of rate control, by means of which the rate control algorithm is informed about the size of the encoded frame. This is necessary for properly calculating the instantaneous state of the transmission buffer, known as VBV - virtual buffer verifier - or CPB - compressed picture buffer, and rate estimation used for a closed-loop rate control algorithm.

III. PROPOSED METHOD FOR DYNAMIC BOOST SCALING

As explained above, the frame boost in the rate control process helps in allocating per frame bits in the GOP and selecting the Qp value. This frame boost calculation in (1) is derived based on empirical data for error per macroblock and there is inefficiency in the projected bits allocation as it is based on recent history. We propose a method to scale the calculated boost by a factor that accounts for the changes in frame complexity over time as shown in (2). This helps prevent the issue of over-boosting the keyframe and golden frame seen in libvpx due to the usage of empirical values for err_per_mb.

$$\text{frame\_boost} = (\text{BSF} * \text{err\_per\_mb} * \text{active\_area}) / \text{inter\_error} \quad (2)$$

As changes to boost are tied to target bits and Qp, which may translate to both objective and subjective poor video quality measure, it is important to have this boost scaling factor (BSF) as a content-dependent correction factor which is updated dynamically.

First-pass statistics in a two-pass encoding can give some information about the coding complexity of each frame. In the first-pass statistics, there are five types of raw data being calculated:

- SSE (sum-of-squared-errors) of intra prediction
- SSE of inter prediction with LAST_FRAME
- SSE of inter prediction with GOLDEN_FRAME
- Block noise energy
- Motion vectors

These raw data are compared with thresholds or directly accumulated to compute 23 statistics at the frame-level, which can later be employed by the rate control algorithm for deciding second pass parameters. In the original libvpx encoder, first-pass data is used for planning allocation of bits for future frames within each GOP interval. In addition to that, the first-pass data can also be used to predict boost scaling. The scaling factor can be computed using a linear or nonlinear prediction model as a function of computed first pass statistics data for the given video frame.

As a machine learning problem to train a prediction model, data samples for each encoded frame (frame sizes and first pass statistics data) are used. These data samples are then classified into bins, according to a frame coding type in VP9, i.e. the frame level in a GF structure (Inter-coded frame, ARF, GF_ARF, KF). The prediction model for the scaling factor was trained separately for each frame type. For each frame type, a number (~10K) of samples for the training data set, and (~10K) samples for the test data set was randomly selected for the purpose of machine learning. Since the number of keyframes (Intra) is relatively small, fewer samples (about 1.5K) were used.

For prediction model for intra frame, 4 parameters relevant to the keyframe complexity estimate from the first pass statistics data were selected for a linear regression model: intra_error (an estimate of per-pixel intra coding error), frame_noise_energy (an estimate of per-block (16x16) noise level), intra_skip_pct, and intra_smooth_pct (both

intra_skip_pct and intra_smooth_pct indicate the percentage of blocks whose intra coding error is less than a threshold, while intra_skip_pct uses a much smaller threshold value). The linear regression parameters or coefficients and the intercept values obtained are in table I.

Table I. Linear Regression Coefficient and Intercept for Intra frames

Regression coefficient	**Value**
intra_error	0.000195704
frame_noise_energy	0.000353872
intra_skip_pct	0.031619777
intra_smooth_pct	-0.628801066
Intercept	0.58201696

For inter frames, the first problem to address is to select the appropriate variables out of the more than 20 first pass statistics since not all variables may have equal significance. Principal Component Analysis (PCA) is used to -

a) determine how many variables (dimensions) are sufficient for the model

b) which of the variables have highest significance for the top principal components (which are eigenvectors of the covariance matrix)

As a result, the following set of 8 variables for the model were selected: sr_coded_error (estimate of per-block inter coding error with GF), frame_noise_energy, pcnt_motion (percentage of blocks coded with last frame), pcnt_second_ref (percentage of blocks coded with GF), pcnt_intra_low, pcnt_intra_high (pcnt_intra_low and pcnt_intra_high are percentage of intra coded blocks with low and high variances, respectively), intra_skip_pct, intra_smooth_pct.

Inter frames are classified by levels, where 0 is a basic inter frame, level 2 is an alternate reference frame (ARF), and level 3 is a GF_ARF frame. A linear regression method with 8 input variables is used to create a prediction model on the training data for GF_ARF which are boosted by libvpx. The linear regression parameters or coefficients and the intercept values obtained are in table II.

Table II. Linear Regression Coefficient and Intercept for Inter frames

Regression coefficient	**Value**
sr_coded_error	0.001651823
frame_noise_energy	0.000128382
pcnt_motion	0.064522889
pcnt_second_ref	-0.029995012
pcnt_intra_low	0.022016676
pcnt_intra_high	-0.04589061
intra_skip_pct	0.003596819
intra_smooth_pct	-0.519506796
Intercept	0.231590335

To explore whether a nonlinear function gives better prediction results for the same training and test data sets, two different powerful nonlinear prediction methods, a Random Forest and a multilayer artificial neural network (ANN), were also tested. Both models are available in Scikit-learn[5], a machine learning library in python, which was used for data analysis and modeling experiments. Random Forest is a supervised learning algorithm, with a group of decision trees, trained with

a bagging method. A multilevel neural network model (ANN) with the same training data was also evaluated with 5 layers, with the following size of each layer 90, 90, 90, 90, 20. However, from a complexity perspective, a linear regression method is preferable. It simplifies implementation, especially for the low-complexity implementation limitations involved.

IV. RESULTS

For our experiment we use the popular Bjontegaard delta bitrate (BD-rate)[6] method with respect to three popular video quality metrics (QM): SSIM[7], PSNR, VMAF[8] for coding efficiency measurement. BD-rate metric gives an average bit-rate reduction (negative values) or increase (positive values) for the same quality between two encoding methods. The command line used in libvpx encoder is given below –

--ivf -o <output_video.vp9> <Input y4m> --codec=vp9 --verbose --passes=2 --limit=300 --i420 --profile=0 --cpu-used=1 --fps=30 --kf-min-dist=150 --kf-max-dist=150 --arnr-maxframes=7 --arnr-strength=5 --lag-in-frames=25 --aq-mode=0 --end-usage=q --cq-level={cq-level} --min-q=0 --max-q=63 --bias-pct=100 --minsection-pct=1 --maxsection-pct=10000 --auto-alt-ref=6 --frame-parallel=0 --threads=1 --tile-columns=0

After implementing the linear regression model discussed above, for both keyframe and golden frame, we wanted to present the impact of our proposed method on the per-frame Qp selection, bits used and QM values. So to show the results, we choose a random KF group of frames (126 frames, 14045 – 14170) for the ElFuente video sequence of 1920x1080 resolution from the Netflix Public dataset[9]. Table III shows the per frame data for the first GF group for this KF group of the video using the baseline libvpx constant quality mode at cq-level=38 and cpu_speed 1. Similarly, table IV shows the same GF group per frame data using our proposed method for the same test conditions. As shown, after the boost scaling, the Qp are adjusted for the boosted frames i.e. frame 0 and 1. The Qp for frame 0 is slightly higher and it reduces the bits used with very small impact on the SSIM and PSNR values. By scaling the boost, we can see the slight improvement in the video quality assessment algorithm - VMAF for frame 0,1 in table IV.

Table III. Per Frame Qp, Bits and QM for Baseline

Frame (14045 - 14059)	Baseline (libvpx)				
	q_index	PKT_SIZE	SSIM	PSNR	VMAF
0	74	31809	0.997603	46.9177	93.4496
1	59	674	0.996559	45.5676	92.4165
2	130	1476	0.996314	45.3841	91.7811
3	137	589	0.996024	45.0727	91.1623
4	141	2701	0.9961	45.0702	91.3431
5	152	923	0.995785	44.7134	90.1095
6	152	545	0.99592	44.8062	90.5713
7	152	5726	0.996174	45.0234	92.2151
8	152	592	0.996015	44.7765	90.96
9	137	1575	0.99609	44.9752	91.515
10	141	600	0.996148	44.9844	91.2507
11	152	2500	0.996381	45.353	92.4232
12	152	943	0.9962	45.2073	92.6589
13	152	602	0.996558	45.6127	94.0201
14	152	55	0.997169	46.434	96.3473

Table IV. Per Frame Qp, Bits and QM for Proposed Method

Frame (14045 - 14059)	Proposed method				
	q_index	PKT_SIZE	SSIM	PSNR	VMAF
0	76	31492	0.997571	46.8912	93.4648
1	71	671	0.996532	45.5365	92.4527
2	130	1403	0.996327	45.3841	91.7575
3	137	590	0.996035	45.0245	91.0466
4	141	2754	0.996093	45.0883	91.4413
5	152	879	0.995859	44.6856	90.1501
6	152	567	0.995897	44.7681	90.6775
7	152	5719	0.996136	44.99	92.2395
8	152	612	0.995943	44.7338	90.6476
9	137	1595	0.995982	44.9167	91.4758
10	141	591	0.996045	44.9149	91.116
11	152	2500	0.99625	45.2643	92.4598
12	152	914	0.996057	45.1139	92.0233
13	152	623	0.996388	45.4832	93.0566
14	152	32	0.996982	46.242	96.0407

To examine the effect of the proposed method, we perform a statistical analysis of the qualities and bitrates achieved by two rate control (baseline and proposed) methods. We first aggregated qualities and bitrates from all 126 frames (14045 – 14170) and calculated the average bitrates and average quality metrics. Finally, we looked at the aggregate changes of bitrate and quality metrics between the default libvpx and our proposed method. In particular, we chose libvpx to serve as the comparison target, and examined the difference of bits, SSIM, PSNR, and VMAF, for our proposed method. Table V shows the results. We can see that the delta for bits spent is negative, i.e., the proposed method uses less bits than the baseline. Although the negative values for changes in quality metrics indicate that they are worse than the baseline, these deltas are very small.

Table V. Average Bits Used, SSIM, PSNR, VMAF Differences

	Δ PKT_SIZE	**Δ SSIM**	**Δ PSNR**	**Δ VMAF**
Libvpx	0	0	0	0
Proposed	-10.881	-6.4E-05	-0.04986	-0.0881

For further evaluation, we used two datasets - Facebook (FB) internal dataset[10] which are 400 top-viewed public videos from FB Pages, and the popular Derf[11] dataset. The FB videos were tested in an anonymized manner without subjective analysis. The cq-level values used are {33, 39, 45, 51}. All comparisons are made with respect to libvpx at the corresponding preset (cpu_speed). The BD-rate results are shown in Tables VI and VII while Table VIII shows the complexity or runtime differences.

Table VI. BD-rate Delta Results for FB Internal dataset

	Bd-rate delta (2p-q)			Bd-rate delta (2p-cq)		
	SSIM	PSNR	VMAF	SSIM	PSNR	VMAF
cpu 1	-0.52%	-1.18%	-1.65%	-1.34%	-2.28%	-2.94%
cpu 2	-0.45%	-1.10%	-1.52%	-1.25%	-2.17%	-2.78%
cpu 4	-0.56%	-1.21%	-1.75%	-1.46%	-2.34%	-2.90%
cpu 7	-0.57%	-1.27%	-1.78%	-1.35%	-2.33%	-2.99%

Table VII. BD-rate Delta Results for DERF Dataset

	Bd-rate delta (2p-q)			Bd-rate delta (2p-cq)		
	SSIM	PSNR	VMAF	SSIM	PSNR	VMAF
cpu 1	-0.63%	-1.52%	-2.60%	-2.14%	-2.78%	-3.29%
cpu 2	-0.72%	-1.64%	-2.87%	-2.02%	-2.75%	-3.25%
cpu 4	-1.02%	-2.00%	-3.33%	-2.02%	-2.74%	-3.26%
cpu 7	-0.92%	-1.93%	-3.03%	-2.29%	-2.91%	-3.45%

Table VIII. Encoding Runtime / Complexity Measurement

	Libvpx	Proposed method	ratio
cpu 1	76.76	76.71	0.999349
cpu 2	62.19	63.32	1.01817
cpu 4	51.18	51.08	0.998046
cpu 7	49	49.9	1.018367

As we can see, there is virtually no difference in runtime/complexity, while coding efficiency has improved across different presets (cpu_speed) and also across two rate control methods – constant quality (2p-q) and constrained quality (2p-cq). Evidently, the proposed method of dynamically scaling of frame boost improves quality and rate control operation, due to better Qp adjustment for the boosted frames in a GOP.

V. CONCLUSION

In this work, we propose a new method to use first-pass statistics as inputs to a prediction model to dynamically adjust the frame boost of the quantization parameter. We can replace the traditional models with better prediction models using a linear model function for the frame boost calculations. Two sets of feature statistics are discovered using the PCA method for key frames and golden frames, respectively. We compare the associated coding performance using linear regression and present the experimental results for typical Facebook video content, which show improved BD-rate performance compared to the original libvpx VP9 encoder. The proposed method helps in preventing the over boosting of key frames and golden frames by accounting for the various characteristics in the content, mainly in user generated content, which is a beneficial feature for all applications. As part of future work, we would like to try simplified implementations for non-linear models to further improve prediction accuracy.

REFERENCES

[1] "libvpx, code repository - open-source VP9 encoder/decoder software," link: https://github.com/webmproject/libvpx.

[2] Mukherjee, Debargha et al., (2015). "A Technical Overview of VP9--the Latest Open-Source Video Codec," SMPTE Motion Imaging Journal. 124. 44-54. 10.5594/j18499.

[3] VP9 Bitstream and Decoding Process Specification, https://www.webmproject.org/vp9/

[4] Jim Bankoski, Paul Wilkins, Yaowu Xu, "Technical Overview of VP8, an open source video codec for the web"

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011

[6] G.Bjøntegaard,"Calculation of average PSNR differences between RD-curves (VCEG-M33)," in VCEG Meeting (ITU-T SG16 Q. 6), 2001.

[7] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP .2003.819861.

[8] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., and Manohara, M., "Toward a practical perceptual video quality metric," https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, 2016.

[9] "https://github.com/Netflix/vmaf/blob/master/resource/doc/datasets.md" - Netflix Public Dataset

[10] Yu Liu, Open Source, Video Engineering, "AV1 beats x264 and libvpx-vp9 in practical use case", https://engineering.fb.com/video-engineering/av1-beats-x264-and-libvpx-vp9-in-practical-use-case/, April 2018

[11] Xiph.org Video Test Media [derf's collection], https://media.xiph.org/video/derf/