Machine learning-based energy consumption modeling and comparing of H.264 and Google VP8 encoders

Yousef O. Sharrab[1], Mohammad Alsmirat[2], Bilal Hawashin[3], Nabil Sarhan[4]
[1,4]Wayne State Multimedia Systems and Deep Learning Research Laboratory, ECE Department, Detroit, USA
[2]Department of Computer Science, Jordan University of Science and Technology, Jordan
[3]Department of Computer Information Systems, Al Zaytoonah University of Jordan, Jordan

Article Info

Article history:

Received Apr 10, 2020
Revised Jun 7, 2020
Accepted Jun 23, 2020

Keywords:

Encoders comparison
Encoding energy consumption
Encoding time
Feature engineering
Google VP8 encoder
Machine learning
MPEG-4 Part 10
H.264/AVC
Perceived video quality
Regression, modeling
Video encoders
Video encoding

ABSTRACT

Advancement of the prediction models used in a variety of fields is a result of the contribution of machine learning approaches. Utilizing such modeling in feature engineering is exceptionally imperative and required. In this research, we show how to utilize machine learning to save time in research experiments, where we save more than five thousand hours of measuring the energy consumption of encoding recordings. Since measuring the energy consumption has got to be done by humans and since we require more than eleven thousand experiments to cover all the combinations of video sequences, video bit_rate, and video encoding settings, we utilize machine learning to model the energy consumption utilizing linear regression. VP8 codec has been offered by Google as a free video encoder in an effort to replace the popular H.264 video encoder standard. This research model energy consumption and describes the major differences between H.264/AVC and VP8 encoders based on of energy consumption and performance through experiments that are machine learning-based modeling. Twenty-nine uncompressed video segments from a standard data-set are used, with several sizes, details, and dynamics, where the frame sizes ranging from QCIF(176x144) to 2160p(3840x2160). For fairness in comparison analysis, we use seven settings in VP8 encoder and fifteen types of tuning in H.264/AVC. The settings cover various video qualities. The performance metrics include video qualities, encoding time, and encoding energy consumption.

Corresponding Author:

Mohammad Alsmirat
Department of Computer Science
Jordan University of Science and technology, Jordan
Email: masmirat@just.edu.jo

1. INTRODUCTION

Machine learning approaches have recently contributed to the advancement of the prediction models used for energy consumption [1–3]. In this paper, we utilize machine learning to predict the energy consumption of over ten thousand experiments to be used for encoders performance comparisons.

The popularity of H.264/AVC has confronted an incredible competence with Google discharging VP8 and VP9 as free of charge video encoders, mainly on YouTube [4]. One of the key factors that would shape the encoder of the future is the adequacy of VP8 comparing to H.264/AVC. Thanks to the awesome overextend of video compression, the viability of VP8 comparing to with H.264/AVC needs more thorough analysis. It is noted that a few works only compared the viability of both encoders such as [5–14]. Study [9] has compared

H.264/AVC and VP8 encoders for only three sequences.The evaluation measurements were only one metric; which is the perceived video quality. Besides, it tested only with minimum basic compression parameters.

This research depicts the contrasts between MPEG-4 Part 10 (H.264/AVC) and VP8 encoders in energy consumption, video quality, and encoding time/speed. It moreover gives nitty-gritty comparative evaluations for more than eleven thousand tests, so as to reflect practical circumstances by wisely selecting the compression settings, and the video evaluation segments. Twenty-nine uncompressed video segments were used, with a variety of frame contents, with sizes extending from QCIF (176x144) to 2160p (3840x2160) and the contents changing significantly in detail and speed of movements. To guarantee a reasonable and fair comparison, we utilize three encoding settings in H.264/AVC, with five different tunings for each, and seven settings in VP8 encoder. These settings cover an assortment of accomplished quality levels. We change the bit_rate for each video segment to study different video quality. The performance metrics include the quality of the video, speed of encoding, and energy consumption. For the quality of the video, we use the *structural similarity index* (SSIM) [15].

The organization of the paper is as takes after. Section 2. provides background information of the used encoders for comparison. Subsequently, Section 3. analyzes and discusses the performance assessment and evaluation procedures, techniques, and methodology. At last, Section 4. illustrate and analyzes the most results.

2. BACKGROUND INFORMATION

Most of the researches that combine encoders and machine learning/deep learning (ML/DL) techniques focused on developing robust ML/DL algorithms for the detection, tracking, and classification of objects [16, 17] and the detection and classification of unusual events [16–21] (and reference within). The overwhelming majority of research on CV considered the development of robust algorithms to improve accuracy [22–26] (and references within).

The encoding process stages of an H.264/AVC codec include spatial(intra) and temporal (inter) prediction [27]. Both the spatial and the temporal prediction are utilized to decrease the duplicates within the video. Video information contains temporal and spatial repetition. Subsequently, similitudes can be encoded by fair considering spatial (within a frame) and/or temporal (between frames) residuals.

An H.264/AVC codec can select from numerous diverse intra-prediction and motion-estimation modes at the time of encoding a macroblock. The rate-distortion optimization (RDO) mode selection is an optimum algorithm for choosing the most excellent encoding mode for each macroblock, based on a cost that considers the lowest value of bit_rate and distortion combination. The differences between H.264 and VP8 in modes are shown in Table 1.

Table 1. H.264/AVC and VP8 encoding comparison

Encoding Step	H.264/AVC encoder	VP8 encoder
Intra-Prediction	Nine modes of prediction were used per (4×4) and (8×8) Luma block in high profiles prediction $four$ modes of prediction were used per (16×16) Luma block and (8×8) Chroma prediction modes	Uses $four$ intra-prediction modes shared (4×4) and (16×16) Luma, and (8×8) Chroma
Inter-Prediction	sixteen reference frames	three reference frames
	Partition types are (16×16) down-to (4×4)	Partition types are (16×16) down-to (4×4)

Spatial-prediction is utilized to foresee the content of a block from its neighbors' blocks with no history utilization. In spatial (intra-mode) selection in H.264/AVC , the number of possible mode combinations for a 16×16 pixel MacroBlock (MB) can be $(16 \times N4 + N16) \times N8$, where $N4$, $N16$, and $N8$ represent the number of modes of a 4×4 Luma block, and a 16×16 Luma block, an 8×8 Chroma block,respectively. To select the best mode for one MacroBlock, the encoder performs $(16 \times 9 + 4) \times 4$ which is equals to 592 RDO calculations [28]. In VP8 codec, applying the same equation gives $(16 \times 4 + 4) \times 4$ which is equals to 272 RDO calculations. Hence, the complexity of H.264/AVC is more than twice that of VP8 encoder intra-prediction without optimized implementation.

Motion-estimation (Inter-prediction) is the method of predicting the a block contents (Luma and Chroma) by referring to past and/or future processed frames. Reference frames (RFs) and motion vectors (MVs) are the essential parts of motion estimation (inter-mode prediction). The RF is already processed frame utilized to get similar blocks, and where the MV show the displacement (in pixels) between where the current block is in the current frame and the comparing prediction block within the RF [29].

H.264/AVC and VP8 encoders have comparative structures in terms of block sizes. H.264/AVC utilises the following partitions 16×16, 16×8, and 8×16, and each 8×8 can be further divided into 8×8, 8×4, 4×8, or 4×4. While VP8 encoder utilises 16×16, 16×8, 8×16, 8×8, and 4×4 partitions. VP8 encoder dropped 8×4 and 4×8 partitions which is impossible to be of a significant issue [30, 29]. Both H.264/AVC and VP8 encoders support variable-size motion vectors.

3. PERFORMANCE EVALUATION METHODOLOGY

3.1. Used codecs, video sequences, and performance metrics

As for the experimental set up, we adopted vpx codec for encoding and decoding with regard to the VP8. Furthermore, we adopted for encoding X264 (r1688) and we adopted for decoding FFmpeg (SUN-r24758) with regard to the H.264/AVC. The evaluation video segments were categorized into 4 categories according to frame size, as illustrated in Table 2. provide more details about the used test sequences are in Table 3.

Table 2. Used video segments properties

formatting	frmae size (in Pixles)	Standard	bit_rate
Quarter common intermediate format (QCIF)	176×144	videoconferencing	20-800 Kbps
Common intermediate format (CIF)	352×288		
4 Common intermediate format (4CIF)	704×576	Standard-Definition Television (SDTV)	100-2000 Kbps
Standard HD (720p)	1280×720	High-definition television (HDTV)	500-3000 Kbps
Full HD (1080p)	1920×1080		
4k (2160p)	3840 × 2160	Quad high-definition television (QHDTV)	2000-8000 Kbps

Table 3. Characteristics of the used standard video sequences

Video Segment	Duration (seconds)	Video Size (Frames)	Resolution
Forman	12	300	176×144, 352×288
Salesman	17	449	176×144
News	12	300	176×144, 352×288
Mobile	12	300	176×144
Highway	80	2000	176×144
Stefan	3	90	352×288
Paris	42	1065	352×288
MotherDaughter	12	300	352×288
City	24	600	704×576
Crew	24	600	704×576
Harbour	24	600	704×576
Soccer	24	600	704×576
Ice	19	480	704×576
DucksTakeOff	20	500	1280×720, 3840 × 2160
InToTree	5	500	1280×720, 3840 × 2160
OldTownCross	5	500	1280×720, 3840 × 2160
ParkJoy	20	500	1280×720, 3840 × 2160
Mobcal	20	504	1280×720
BlueSky	8	645	1920×1080
Pedestrian Area	15	375	1920×1080
Riverbed	10	250	1920×1080
RushHour	20	500	1920×1080
Station2	12	313	1920×1080

The used evaluation measurements are energy consumption, video quality, and encoding speed performance. To measure the quality of the videos , we utilize the *structural similarity index* (SSIM) [15]. To determine SSIM, we utilized the the brightness in the image or Luma (Y) part in the YUV, as brightness would stimulate eye more than the color would. We split the energy measured data to two categories, training data and test data (80% of measured data were for training).

3.2. Procedure

Encoding is done using vpxenc/vpxdec (http://www.webmproject.org) codec. H.264/AVC encoding is done by X.264 (http://X264.nl), while the decoding of H.264/AVC is done by using FFmpeg. We developed C++ program to computerize the measurements of the encoding time and the encoded video bit_rate.

We utilize MATLAB program to measure SSIM metrics, to find the quality of the distorted decoded video sequences relative to the original distortion-free video sequence. The ratio of the target bit_rate to the accomplished bit_rate is calculated by dividing the input bit_rate that we specify in the command line by the accomplished bit_rate, which is the average of the real bit_rate for the encoded sequences. To tradeoff the encoding speed and the accomplished bit_rate for certain SSIM quality, we draw the point which corresponds to the interpolation tuple of that SSIM quality. In these figures, the higher speed and lower bit_rate are better encoders and parameters combinations.

The encoding settings for both H.264/AVC(X264) and VP8 encoders are selected to include a variety of energy consumption and video quality. The encoding settings of VP8 encoder are selected from the WebM website. Additionally, H.264/AVC (X264) parameter settings are selected from the developers website [8]. The utilized encoding settings are explained in [11]. The formats of the encoded videos are WebM and mp4 for VP8 encoder and H.264/AVC, respectively.

The energy consumption experiments were conducted on a dual-core processor laptop. The energy was measured by Graphic Timer Watt meter. The number of needed experiments are (29 video sequence $\times$ 6 bitrates $\times$ 22 settings $\times$ 3 repetition for accuracy which is equals to $11,484$). Each experiment needs in average half an hour to be setup and conducted manually, which is very time consuming and prone to error. For that reason, we pick a sample as follows, for each sequence we measure the energy for one bit_rate out of six, and 13 settings out of 22 which reduces the number of needed experiments to 522 experiments ($29 \times 1 \times 6 \times 3$ repetitions for accuracy). We use this collected data for training, to model the energy consumption based on the encoding time feature. From the developed model, we predicted the remaining $10,962$ $(11,484 - 522)$ energy consumption predictions. The block diagram of the energy consumption modeling including training and testing is shown in Figure 1.

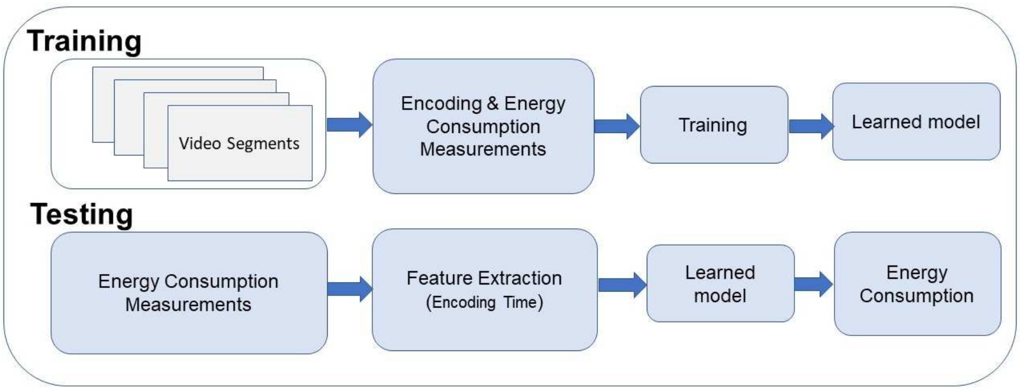

Figure 1. Machine learning-based energy consumption modeling block diagram

4. RESULT PRESENTATION AND ANALYSIS

In this section, we analyze the results of H.264/AVC and VP8 encoders Comparison in energy consumption, encoding time, and video quality. For space, we do not show all figures. For clear figures, we do not show all setting results in all figures.

4.1. Energy consumption and encoding speed against perceptual quality at certain bit_rate tradeoff

In all frame sizes excluding *high definition standards* (1080p and 2160p), H.264/AVC accomplishes lower energy consumption and higher performance than VP8 encoder at the same bitrate. In High Definition Standards resolutions, VP8 encoder yields higher performance but more encoding time and higher energy consumption than H.264/AVC at the same performance. Notes that we use different settings for *high definition standards* (720p, 1080p, and 2160p) for H.264/AVC based on the developers recommendations. Upon the evaluation of VP8 encoder settings under QCIF, CIF, and 4CIF, with "Best" indicating the most efficient performance, the maximum energy consumption, and the slowest speed of encoding, settings "Good 0", "Good 1", "Good 2", "Good 3", "Good 4", and "Good 5" come next,respectively. The setting "Good 5" produces the worst performance, the minimum energy intake, and the fastest encoding among all the VP8 encoder settings. On the other hand, settings "Good 1", "Good 2", and "Good 3" (720p, 1080p, and 2160p) were more efficient in terms of encoding speed, performance, and energy intake. Regarding H.264/AVC, the setting "higher quality" has clearly the slowest encoding, the maximum energy intake, and the most efficient performance, mainly for smaller resolution values, whereas adopting "Higher Speed" provides the worst quality, the smallest energy intake, and the fastest encoding. "Normal quality" is in between the previous two settings.

4.2. Performance at various bit_rate

Figure 2 compares the H.264/AVC and VP8 encoding time (speed) and video quality performance P according to the following model

$$P = a \times Q/Q_h + (1 - a) \times S/S_h, \quad (1)$$

where a is weighting factor, Q, Q_h, S, S_h represents quality, highest quality, encoding speed, highest encoding speed, respectively. The weighting factor take values fro 0 to 1, its value can be set based on which we care more about; the quality or the encoding speed. For example in real time streaming we care more about encoding speed, while in off line recording we care more about the quality. This model takes the mean value of the normalized perceived quality and encoding speed for $a = 0.5$. The figure shows that 'VP8 Good 5 setting' and 'H.264/AVC High Speed setting' have the best performance, where 'VP8 Best setting' and 'H.264/AVC High Quality setting' have the worst performance. This evaluation shows that high encoding speed (less quality) settings perform better as they increase the encoding speed without significant impact on the perceived video quality. H.264/AVC encoder generally has higher performance than Vp8 encoder based on (1).

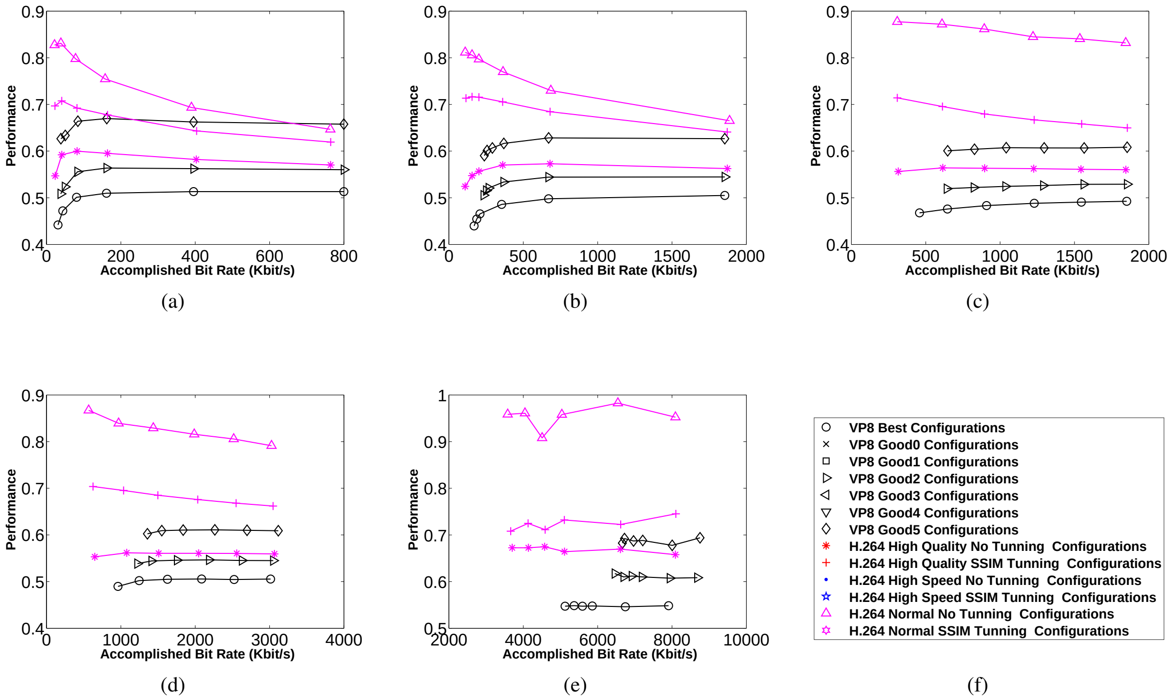

Figure 2. Codecs comparison in performance=(0.5× quality/highest quality +0.5× encoding speed/highest encoding speed) Vs. bit_rate, (a) Quality QCIF and CIF, (b) Quality 4CIF, (c) Quality 720p, (d) Quality 1080p, (e) Quality 2160p

As we see in Figures 3 and 4, H.264/AVC generally consumes less energy than VP8 encoder, only VP8 “Good 5” setting consumes less energy than H.264/AVC “High Quality” setting.

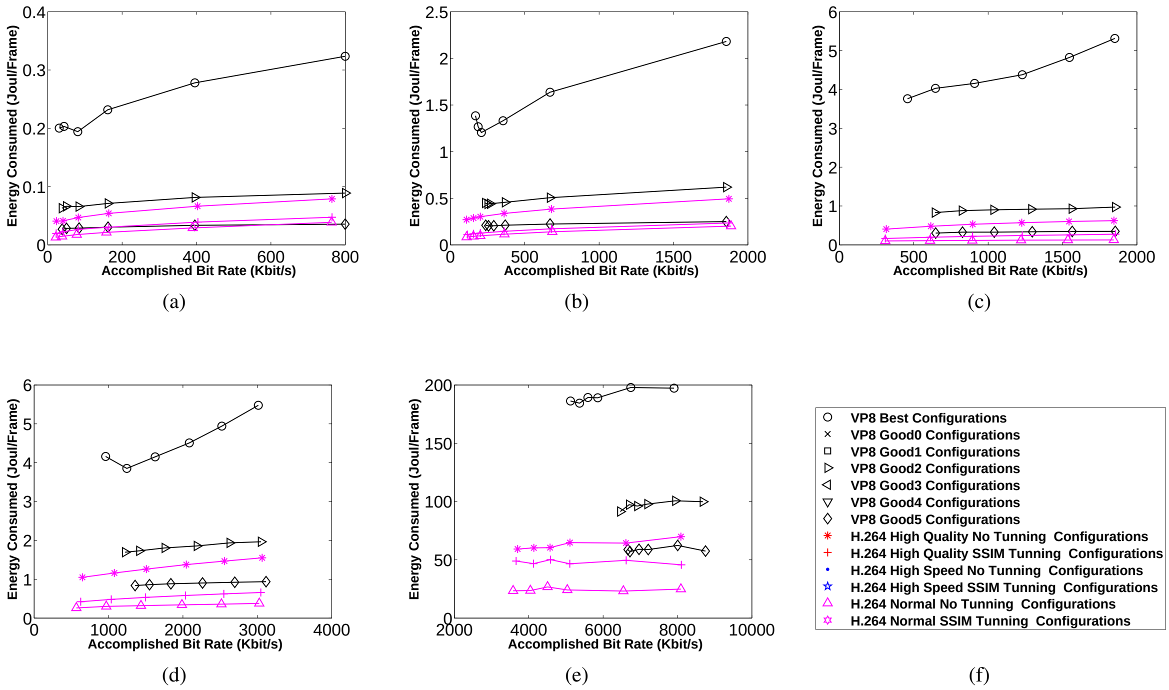

Figure 3. Codecs comparison in encoding energy consumption, (a) Quality, QCIF and CIF, (b) Quality 4CIF, (c) Quality 720p, (d) Quality 1080p, (e) Quality 2160p

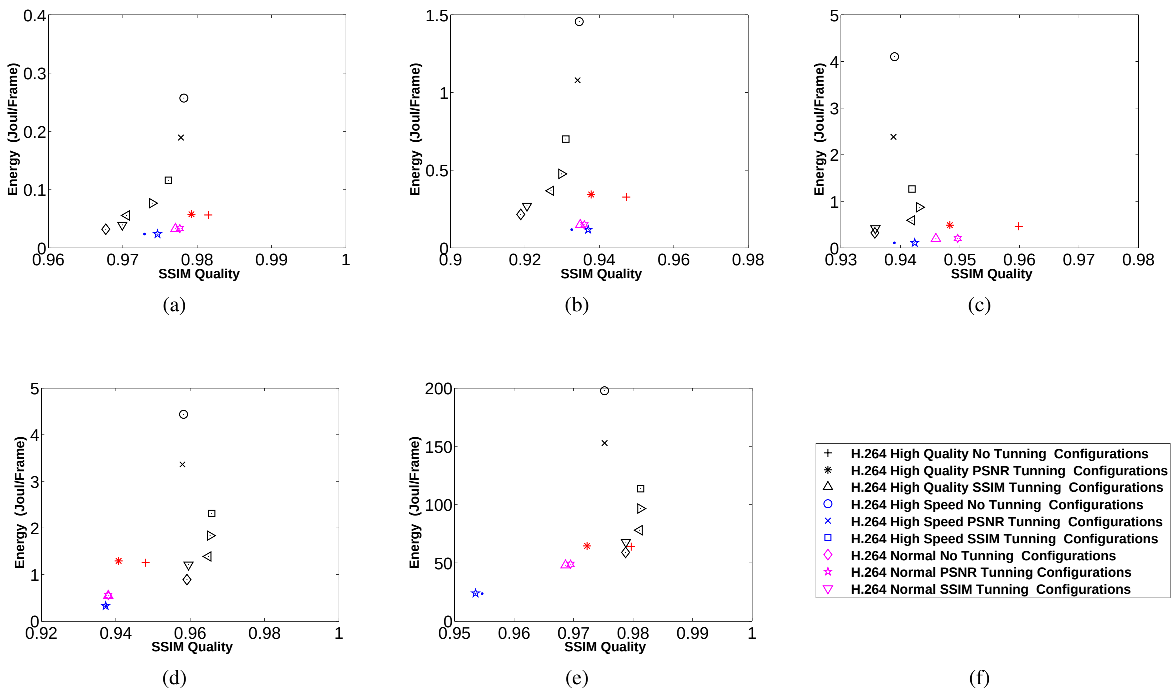

Figure 4. Codecs comparison in encoding energy consumption and quality at settled Bit_rate, (a) Quality, Bitrate=300 Kbps, QCIF and CIF, (b) Quality, Bitrate=500 Kbps, 4CIF, (c) Quality, Bitrate=800 Kbps, 720p, (d) Quality, Bitrate=2000 Kbps, 1080p, (e) QQuality, Bitrate=7000 Kbps, 2160p

5. CONCLUSIONS

Progression of the prediction models utilized in a variety of areas is a result of the contribution of machine learning approaches. We show that we can extend our experiments and have more data to compare by predicting energy consumption based on a machine learning model. The model saved us more than five thousand hours of energy measurements. It is exceptionally imperative to have a way of preparing data through a machine learning model to save time and it is immune to error. Implementation computation complexities have been compared of H.264/AVC and VP8 encoders, and proved that VP8 encoder prediction is more straightforward. Such easiness of prediction results in speedier and lowers energy encoding utilization at the encoder. which can simplify the encoding by an efficient implementation. Furthermore, we have compared and analyzed the execution of H.264/AVC and VP8 encoding through broad tests. In terms of the energy consumption, H.264/AVC is predominant to VP8 encoder for most frame sizes. The results illustrate that H.264/AVC by and large accomplish superior to VP8 Encoder in terms of performance and energy utilization.

REFERENCES

[1] Amir Mosavi and Abdullah Bahmani, "Energy consumption prediction using machine learning: A review," *Qut ePrints*, 2019, https://doi.org/10.20944/preprints201903.0131.v1.

[2] Yousef O. Sharrab and Nabil J. Sarhan, "Aggregate power consumption modeling of live video streaming systems," *Proceedings of the ACM Multimedia Systems Conference*, pp. 60-71, 2013, https://doi.org/10.1145/2483977.2483983.

[3] Yaser Jararweh, Manar Bani Issa, Mustafa Daraghmeh, Mahmoud Al-Ayyoub, and Mohammad A Alsmirat, "Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 262–274, 2018, https://doi.org/10.1016/j.suscom.2018.07.005.

[4] Holmes Wilson, " Open letter to google: Free vp8, and use it on youtube," 2010.

[5] Chouitek Mama, Benouzza Noureddine, and Bekkouche Benaissa, "Control of variable reluctance machine (8/6) by artificiel intelligence techniques," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 1893-1904, 2020, https://doi.org/10.11591/ijece.v10i2.pp1893-1904.

[6] Pei Xie, Hong Zhang, Weike You, Xianfeng Zhao, Jianchang Yu, and Yi Ma, "Adaptive vp8 steganography based on deblocking filtering," *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pp. 25-30, 2019, https://doi.org/10.1145/3335203.3335711.

[7] Mauricio Altamirano Silva, Raul Bertone, and Jorg Schafer, "Topology distribution for video-conferencing applica- tions," *IEEE 10th International Conference on Networks of the Future (NoF)*, pp. 90–97, 2019, https://doi.org/10.1109/NoF47743.2019.9014957.

[8] Dmitriy Vatolin, Dmitriy Kulikov, and Alexander Parshin, "Sixth MPEG-4 AVC/ H.264 video codecs comparison-short version," 2010.

[9] Patrick Seeling, Frank H.P. Fitzek, Gergo Ertli, Akshay Pulipaka, and Martin Reisslein. "Video network traffic and quality comparison of vp8 and H.264 svc," *Proceedings of the 3rd workshop on Mobile video delivery. Conf. on Human Factors in Computing Systems*, pp. 33-38, 2010, https://doi.org/10.1145/1878022.1878031.

[10] Christian Feller, Jurgen Wunschmann, Thorsten Roll, and Albrecht Rothermel, "The vp8 video codec - overview and comparison to h.264/avc," *IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)*, pp. 57–61, 2011, https://doi.org/10.1109/ICCE-Berlin.2011.6031852.

[11] Yousef O. Sharrab and Nabil J. Sarhan, "Detailed comparative analysis of VP8 and H.264," *IEEE International Symposium on Multimedia*, pp. 133–140, 2012, https://doi.org/10.1109/ISM.2012.33.

[12] Yousef O. Sharrab and Nabil Sarhan, "Accuracy and power consumption tradeoffs in video rate adaptation for computer vision applications," *IEEE International Conference on Multimedia and Expo*, pp. 410–415, 2012, https://doi.org/10.1109/ICME.2012.77.

[13] Muhammad Shafique, Bastian Molkenthin, and Jorg Henkel, "An HVS-based adaptive computational complexity reduction scheme for H.264/AVC video encoder using prognostic early mode exclusion," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 1713–1718, 2010, https://doi.org/10.1109/DATE.2010.5457091.

[14] Yousef O Sharrab and Nabil Sarhan, "Modeling and analysis of power consumption in live video streaming systems," *ACM Transactions on Multimedia Computing, Communications, and Applications*

(TOMM), vol. 13, no. 4, pp. 1–25, 2017, https://doi.org/10.1145/3115505.

[15] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 4, pp. 600–612, 2004.

[16] Omar Javed and Mubarak Shah, "Tracking and object classification for automated surveillance," *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pp. 343–357, 2002.

[17] Xiaojing Yuan, Zehang Sun, Yaakov Varol, and George Bebis, "A distributed visual surveillance system," *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 199-204, 2003, https://doi.org/10.1109/AVSS.2003.1217922.

[18] W. Niu and J. Long, "Human activity detection and recognition for video surveillance," *Proceedings of the IEEE Multimedia and Expo Conference (ICME)*, pp. 719–722, 2004, https://doi.org/10.1109/ICME.2004.1394293.

[19] Robert Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, and Osamu Hasegawa, "A system for video surveillance and monitoring,"*Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA*, pp. 1-68, 2000.

[20] Mohammad Alsmirat and Nabil J Sarhan, "Cross-layer optimization for automated video surveillance," *IEEE International Symposium on Multimedia (ISM)*, pp. 243–246, 2016, https://doi.org/10.1109/ISM.2016.0055.

[21] Mahmoud Al-Ayyoub, Nusaiba Al-Mnayyis, Mohammad A Alsmirat, Khaled Alawneh, Yaser Jararweh, and Brij B Gupta, "Sift based roi extraction for lumbar disk herniation cad system from mri axial scans," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-9, 2018.

[22] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[24] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 1, 2017.

[25] Waqas Sultani, Chen Chen, and Mubarak Shah, " Real-world anomaly detection in surveillance videos," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[26] Lukas Esterle and Peter R Lewis, "Online multi-object k-coverage with mobile smart cameras," *Proceedings of the 11th International Conference on Distributed Smart Cameras (ICDSC)*, pp. 107–112, 2017.

[27] Iain E. G. Richardson. The H.264 Advanced Video Compression Standrad, second edition, WILEY, *Published Online. Vcodex Limited, UK*, 2010.

[28] Jongho Kim, Donghyung Kim, and Jechang Jeong, "Complexity reduction algorithm for intra mode selection in H.264/AVC video coding," *Advanced Concepts for Intelligent Vision Systems (ACIVS'06)*, vol. 4179, pp. 454–465, 2006.

[29] Jason Garrett-Glaser, "The first in-depth technical analysis of VP8," *http://x264dev.multimedia.cx/archives/377.*

[30] Jim Bankoski, Paul Wilkins, and Yaowu Xu, "Technical overview of VP8, an open source video codec for the web," *IEEE International Conference on Multimedia and Expo*, pp. 1-6, 2011.