2023 26th International Conference on Computer and Information Technology (ICCIT) 13-15 December, Cox's Bazar, Bangladesh

Time Complexity Reduction of CU Splitting in HEVC Intra Coding Using Zerocross Edge Detection Method

Md. Zahirul Islam, Tanvir Ahmed Redoy, Mansura Naznine, Mastura Mohenee,
Rizoan Toufiq, Mir Safwan Marzouq, Md Jawad Bin Khaleque

km.zahir61@gmail.com, redoyta19@gmail.com, naznine31@gmail.com, moheneemastura@gmail.com,
rizoantoufiq@gmail.com, marzouq4565@gmail.com, jawaddhew98@gmail.com

Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology, Bangladesh

***Abstract*—Video compression is a crucial technique for efficiently storing and transmitting videos over networks. The prevalent High Efficiency Video Coding (HEVC/H.265) standard achieves significant bitrate reduction but at the cost of increased time complexity. HEVC relies on coding tree unit (CTU) partitioning and Rate Distortion Optimization (RDO) during this process, which involves extensive rate-distortion trade-off searches. Determining the optimal partition for each Coding Unit (CU) within video frames poses a significant complexity challenge given the multitude of combinations to evaluate. As the number of CUs and partitioning grows, so does computational complexity. To address this, we propose an innovative heuristic model based on edge detection within HEVC's intra-mode. Our model employs the Zerocross edge detection algorithm identifying edges in each frame by detecting zero crossings in the second derivative of the image. Experiments show that up to 49.49% reduction in encoding time for intra-mode HEVC across 12 commonly used test video sequences is possible when implementing our approach and omitting the RDO process. However, this efficiency gain comes with an average 3.35% bit-rate increase compared to HEVC.**

***Index Terms*—High Efficiency Video Coding (HEVC), Rate Distortion Optimization (RDO), Edge Detection, Coding Tree Unit (CTU) Partition, Encoding Time Reduction.**

I. Introduction

Video compression, a fundamental process in multimedia engineering, aims to efficiently reduce the data size of digital video. It involves the utilization of encoding and decoding components collectively referred to as a CODEC. This operation strategically addresses spatial redundancy, focusing on similarities among pixels within a single frame, and temporal redundancy, addressing similarities across consecutive frames [1]. Through this optimization of data efficiency, compression leverages these redundancies while maintaining video integrity and content quality [2].

Following the introduction of the Advanced Video Coding (AVC) standard, HEVC represents a substantial breakthrough in the realm of video compression. HEVC has demonstrated remarkable success by achieving approximately 50% reduction in bit rates compared to its predecessors, making it well-suited for the demands of modern multimedia applications [3]. Notably, HEVC excels in efficiently compressing high-resolution videos, boasting enhanced motion compensation and adaptable partitioning of coding tree units (CTUs) [4]. The utilization of the Rate-Distortion Optimization (RDO) technique contributes to the outstanding encoding efficiency. However, this significant efficiency enhancement is offset by a notable surge in computational complexity [5]. The substantial challenge posed by this heightened computational complexity profoundly impacts the development of cost-effective multimedia systems [6]. Therefore, the imperative task is to mitigate the encoding complexity of H.265 to a manageable level.

This research is fundamentally centered on optimizing the CTU partitioning process, seeking a more efficient alternative to the RDO procedure with the aim of reducing time complexity. Prior investigations in this domain have explored solutions for both HEVC intra and inter modes, offering various recommendations. Previous works have encompassed machine learning-based and heuristic models to tackle time complexity challenges. In the machine learning realm, Support Vector Machines (SVMs), Decision Trees (DTs), and diverse deep learning models have been harnessed. A notable example by Heindel et al. [5] delved into SVMs for coding decisions within the HEVC inter coding process. Their approach incorporates six distinct features as input for the SVM classifiers. Werda et al. introduced another approach to reduce HEVC intra coding complexity by employing a fast CU partition module based on SVM and a Gradient-based intra-prediction mode [7]. The study by N. Westland et al. involved a DT classifier employing seven features, resulting in a 42.1% reduction in encoding time and a 0.7% bit rate increase [8]. Furthermore, an investigation utilizing DT for AV1 encoding showcased a 23.6% reduction in time complexity and a 0.73% bit rate increase [9]. A classification method for categorizing video frame blocks based on single or multiple motions was studied in [10]. The HIDR-CNN approach introduced a convolutional neural network architecture to enhance decision-making within the HEVC intra depth range, resulting in a 27.54% reduction in encoding time, albeit with a marginal 0.99% bit rate increase [11]. The study conducted by Feng et al. [12] proposed a fast HEVC intra coding block partitioning

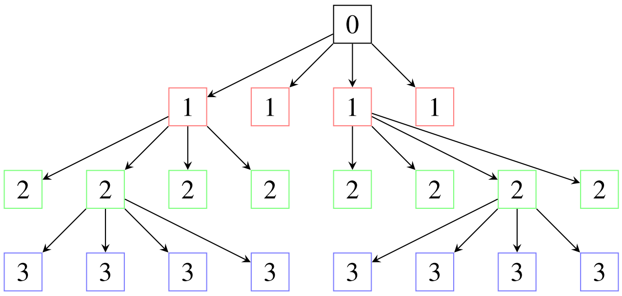

Fig. 1. In the provided figure, a QT (Quad Tree) partition is displayed along with the depth levels. The nodes marked with 0 represent blocks with a size of 64 × 64 (root node). The nodes marked with 1 represent blocks with a size of 32 × 32, corresponding to depth 1. Similarly, the nodes marked with 2 represent blocks with a size of 16 × 16, corresponding to depth 2. Lastly, the nodes marked with 3 represent blocks with a size of 8 × 8, which are at depth 3 (leaf node).

algorithm using Convolutional Neural Network (CNN) based depth map prediction. Again, Hari et al. applied a CNN for CTU partition prediction in intra-mode HEVC [13].

Heuristic methods offer a rapid statistical approach to determining Coding Unit (CU) partitioning. In [14], the authors introduced an efficient CU partitioning technique referred to as the area percent threshold. Experimental results demonstrated that this approach incured only a slight average Bjøntegaard Difference-rate (BD-rate) loss of 0.46%. However, it yields a significant average reduction in encoding time, exceeding 30% [14]. Kim et al. proposed an adaptive keypoint-based CU depth decision (AKCD) system, extending the keypoint-based CU depth decision (KCD) method for assessing CU splitting [15]. Jamali et al. presented a rapid HEVC intra mode decision technique based on edge and the sum of absolute transformed differences costs, resulting in a time reduction of up to 39.2% [16]. Byungjin et al. devised an approach to reduce the selected mode list after the rough mode decision (RMD) for intra coding. They employed a threshold value determined using weights based on the standard deviation of pixel values. The reduced candidate mode was subsequently used in the RDO procedure [17]. BenHajyoussef et al. focused on gradient information in their work, utilizing the Prewitt operator during the pre-processing stage. Gradient values were analyzed to determine the correct mode direction for intra prediction in HEVC [18]. Huang et al. suggested an integrated approach that merges heuristic algorithms and machine learning in [19], introducing the Heuristic Model Oriented Framework (HMOF). This framework includes advanced acceleration algorithms Border Considered CNN (BC-CNN) for CU partition and Naive Bayes for Prediction Unit (PU) partition. The study in [20] proposed a technique for HEVC encoding that estimates RD costs from transformed coefficients. It introduces Sum of Absolute Transform Differences (SATD) methods based on e Discrete Cosine Transformation (DCT) and Discrete Sine Transformation (DST). Menon et al. introduced the Intra CU Depth Prediction (INCEPT) algorithm, leveraging a DCT energy-based feature to limit the RDO process [21]. To reduce computational complexity, heuristic approaches offer several advantages over learning-based methods. They are computationally efficient and require no training overhead, resulting in decreased latency, predictable resource consumption, and scalability. Moreover, heuristic methods do not require online learning, can be parallelized, and provide computational performance stability. In addition, these methods can reduce energy consumption in devices with limited resources. These benefits make heuristic methods a valuable choice when computational efficiency is crucial.

The authors of this research introduced a heuristic approach for HEVC intra-coding. Traditional video codecs rely on the computationally intensive RDO process, which involves extensive calculations. In contrast, their proposed solution offers a faster and more efficient means of determining optimal CTU partitioning within video frames. This method utilizes systematic block division and the zero-crossing technique to identify objects in video frames, enhancing the video encoding process. Overall, this heuristic approach significantly accelerates the procedure, making it particularly well-suited for real-time applications.

The remaining portion of the document is organized as follows: In section II, the HEVC CTU partitioning and RDO procedure are explained. Within Section III, two subsections delineate the structural framework of our proposal and modifications to the RDO process aimed at time-saving. The empirical analysis is showcased in Section IV. Lastly, Section V encapsulates the paper's final remarks.

II. HEVC CTU Partition and RDO Process

To increase encoding efficiency, modern video encoders use a block-based framework for motion compensation. The CTU denotes a rectangular block within a video frame and has a size range of 8 × 8 to 64 × 64 pixels [22], [23]. There are two options for a CTU: it can either contain a single CU or subdivide recursively into smaller CUs. To divide a CTU, HEVC uses a quad-tree structure [5], where the root node of the quad-tree represents the full block and has a depth value of 0. The nodes of the tree can either contain four child nodes or none. When the tree is traversed, the nodes' depth rises by 1, for example, 64×64 corresponds to depth 0 and 8×8 to depth 3 [22]. An example of a CTU quad-tree partitioning is shown in Fig. 1. Due to the adaptability of this CTU partitioning structure, the encoding algorithm choose the best partition scheme for each CTU based on the properties of the video material. This adaptability improves coding performance and compression efficiency, leading to superior bit-rate and quality results.

During the CTU partitioning, to determine whether a CU has to be partitioned or not, conventional codecs use Rate Distortion Optimization (RDO) process. RDO is the key to achieving the highest level of compression effectiveness while maintaining the authenticity of visual content [24]. RDO process is based on a brute-force approach. It uses Lagrange rate-distortion cost function to determine cost for all possible combinations of a CU partition and choose the configuration

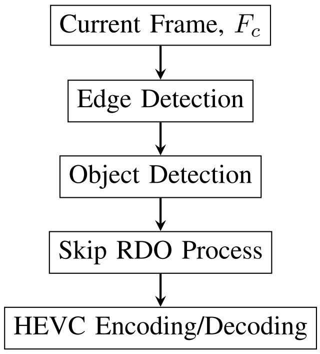

Fig. 2. Block diagram of proposed time complexity reduction approach for HEVC intra-mode coding.

which is best for the rate-distortion trade-off overall. The Lagrange rate-distortion cost function is defined as:

$$\min_{m} J(m) = \lambda.R(m) + D(m) \tag{1}$$

The function depends on a wide variety of parameter combinations (m) and the influential Lagrange multiplier (λ). These parameter settings play a crucial role in adjusting the delicate balance between distortion and bit rate, which in turn affects the size and visual quality of the encoded video. The primary objective is to achieve an appropriate balance where video quality is maintained without excessive data volume growth. Due to the many permutations, especially for critical video sequences, navigating this complex terrain is difficult. For instance, according to [24], even a root-level CU with 64×64 dimensions predicts the occurrence of almost 75,000 different parameter combinations. Increased processing demands are a natural consequence of this complex computational environment, which in turn results in an extensive computational complexity [25]. Additionally, this computational complexity places a significant burden on devices with limited power, particularly android smartphones, and can occasionally result in hardware-related issues. Because of this, it is essential to investigate cutting-edge tactics for accelerating and streamlining the complexities involved with this procedure.

III. Proposed Methodology

The next subsections provide a detailed analysis of the structure and differences between the suggested method and currently used technique. The proposed system framework is described in Section III-A, and the complexity reduction strategy by skipping RDO process in HEVC intra-coding is discussed in Section III-B.

A. Proposed System Framework

The proposed method is shown in Fig. 2. First, as illustrated in the diagram, our system receives the current frame as an input, denoted as F_c. Next, we run this current frame through an edge detection procedure. Then, using the locations of strong edges, we can detect the presence of objects. Finally, we entirely omit the standard HEVC RDO procedure from encoding. In this study, we used the Zerocross edge detection technique.

The zero-crossing detector is a technique used in image processing to locate points in an image's Laplacian where the value transitions through zero, specifically where the Laplacian changes its sign [26]. The image's first-order derivative is computed in the first step. Convolution and a gradient operator are used for this. This operation's output highlights edges and areas of sudden intensity shift. The fundamental formula of the first-order derivative of F_c(x) in x direction is given as:

$$\frac{\partial F_c}{\partial x} = F_c(x+1) - F_c(x) \tag{2}$$

Finding zero-crossings is next after obtaining the gradient image. When the gradient shifts its sign, moving from positive to negative or vice versa, it crosses the zero line. These zero-crossings indicate to the image's edge locations. The second derivative is calculated in x direction is calculated as:

$$Edge(x) = \frac{\partial^2 F_c}{\partial x^2} = F_c(x+1) + F_c(x-1) - 2F_c(x) \tag{3}$$

Similar strategy is adopted in y direction. After finding zero-crossings, the image undergoes to a thresholding step in order to generate a binary edge map. By creating closed contours from zero crossings, this technique produces a binary image with thin edge lines. Zerocross method has benefits like noise robustness, smaller and localized edges, improved augmentation, and increased edge connectivity. The edge position of frame number 10 from the $BQSquare$ sequence is depicted using the zerocross approach in Fig. 3. Fig. 4 illustrates the CTU partition based on strong edge positions.

B. Skipping RDO Process and Time Complexity Reduction

We suggest a new approach to CTU partitioning based on zerocross edge detection in place of the conventional RDO process. This strategy comprises a number of related phases. At first, we process each video frame that makes up a video. The video frame goes through edge detection using the zerocross approach. A number of non-overlapping squares, each measuring 64×64, partition the entire frame. We determine the summation energy for each N×N block as follows:

$$Energy = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} E(x,y) \tag{4}$$

This block is indicated for further division if the summation energy is more than zero, suggesting that there has been objects in it. This block is further divided into four 32×32 blocks. However, further division is avoided if the summing energy is zero. Our edge detection method is extremely effective and has little effect on finding optimal CTU partitioning time. We altered the HM encoder and entirely skipped the traditional RDO procedure in the HM reference program. This method lessens the Complexity involved with CTU partitioning in HEVC.

IV. Experimental Analysis

The following two subsections focus on the experimental analysis part. Section IV-A describes the experimental setup and section IV-B refers to performance analysis.

Fig. 3. Visualization of edge detection (frame number 10) of BQSquare (240p) sequence. Left sided figure is the frame of the sequence and right sided figure is the edges of the frame produced by Zerocross method.

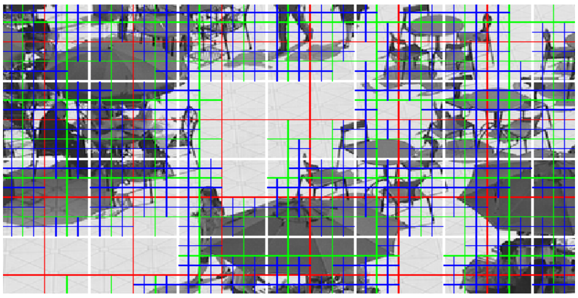

Fig. 4. Visualization of CTU partition (frame number 10) of BQSquare (240p) sequence produces by Zerocross method. The size of white, red, green and blue marked CUs is 64×64, 32×32, 16×16, 8×8 respectively.

TABLE I
ENCODING RESULT: BQSQUARE SEQ. USING HM - 16.20 (UNMODIFIED VERSION) EXPRESSED IN PERCENTAGE (%)

Performance — Qp	**22**	**27**	**32**	**37**
$PSNR$	42.16	38.07	34.50	30.98
$BitRate$	13070.59	8587.75	5516.42	3423.42
$Enc.Time$	317.34	284.58	261.57	238.67

TABLE II
ENCODING RESULT: BQSQUARE SEQ. USING HM - 16.20 (PROPOSED VERSION) EXPRESSED IN PERCENTAGE (%)

Performance — Qp	**22**	**27**	**32**	**37**
$PSNR$	42.09	38.04	34.49	30.98
$BitRate$	13111.41	8594.27	5530.09	3450.68
$Enc.Time$	174.22	155.82	141.56	132.22

A. *Experimental Setup*

We have selected twelve test video sequences approved by the Joint Collaborative Team on Video Coding (JCT-VC) as established standard samples [27]. HM - 16.20 [28] was used to implement the proposed techniques. Four different Quantization Parameters (QP) values— 22, 27, 32, and 37 were used to encode each sequence throughout the experimental study. As our approach is focused on the reduction of complexity in the intra-coding phase of the HEVC, we used $encoder\_intra\_main.cfg$ [27] configuration file to utilize all intra mode.

B. *Performance Analysis*

We used a certain procedure to assess the performance. Each sequence's first 101 frames were used for encoding. At first, We encoded the frames of each sequence using HM - 16.20 (proposed version) [28] and preserve the results for further use. Then we used the proposed HM - 16.20 (modified version) encoder to encrypt the same 101 frames. Table I and Table II are showing the output of all encoding results of the sequence $BQSquare$. After encoding, we used the Bjøntrgaard delta bitrates piece-wise cubic interpolation method [29] to compare the resulting bit rates. The delta encoding time (ΔET) was calculated as follows:

$$\Delta ET = \frac{1}{|S|} \times \sum_{QP_i \in S} \frac{T_1\ (QP_i) - T_2\ (QP_i)}{T_1\ (QP_i)} \tag{5}$$

In the equation, S stands for a set of unique QP values. $T_1(QP_i)$ denotes the amount of time the HM encoder needs to use the i-th QP value to encode a sequence, whereas $T_2(QP_i)$ denotes the amount of time the modified HM encoder needs to use the same i-th QP value to encode the same sequence. Table III is showing summary of all encoding results . Fig. 5 displays the CTU partition of the original HM encoder versus the modified HM encoder employing zerocross edge detection methods. Additionally, Fig. 6 shows the decoded frames for both the original and modified HM which illustrates that there is no detectable subjective difference between those that can be seen with the human eye.

1) **Analysis of Time Complexity Reduction:** The results presented in Table III, demonstrate that our edge detection-based method significantly reduces the time complexity of the original HM implementation. Across a variety of sequences, our method consistently results in substantial time reductions. Notably, when employing the Zerocross edge detection method, as indicated by the Cactus sequence (Class B), we accomplish a remarkable time complexity reduction rate of 49.49%. Our method reduces time complexity by an average of 44.54%, outperforming the method [11] in terms of time complexity reduction. The proposed model is based on heuristic approach which is computationally efficient and doesn't need time for training, validation, and testing phases. Conversely, the model introduced in [11] represents a fast intra prediction algorithm designed for CTU depth range prediction. This model utilized CNN architecture to evaluate rate distortion cost

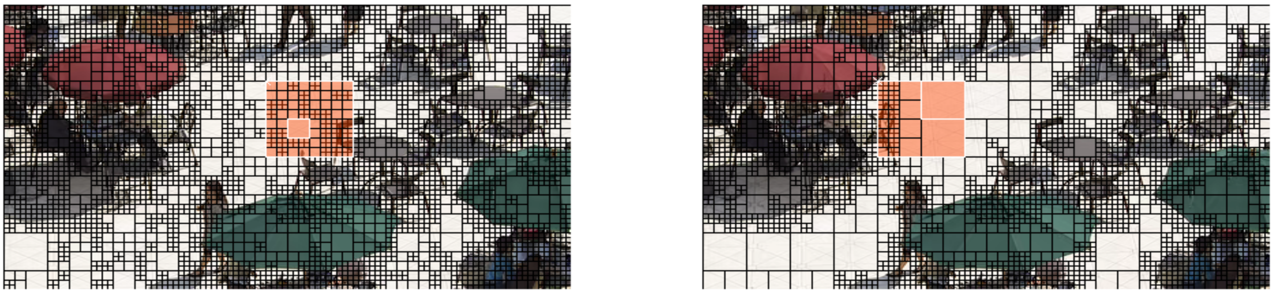

Fig. 5. Visualization of CTU partition (frame number 10) of BQSquare (240p) sequence. Left sided frame is partitioned by HM encoder and right sided frame is partitioned using Zerocross edge detection method.

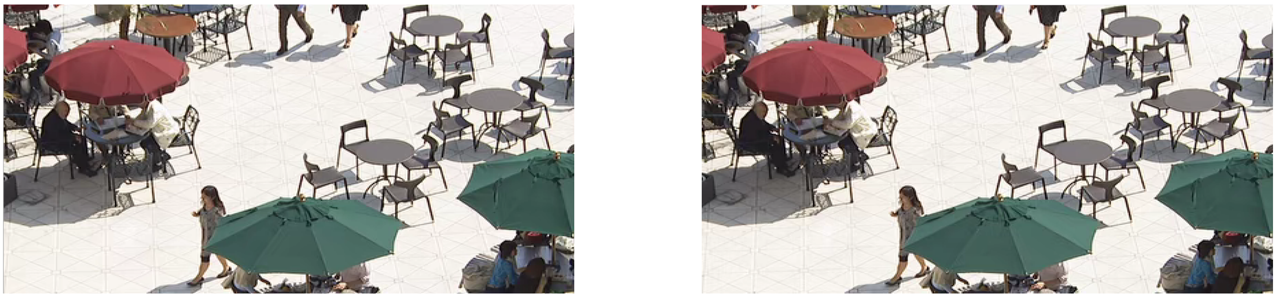

Fig. 6. Visualization of decoded frame (frame number 10) of BQSquare (240p) sequence. Left sided frame is decoded by HM encoder, number of bits = 13070.59, Y-PSNR = 42.16 dB, time = 317.34 sec, QP = 22. Right sided frame is decoded using Zerocross method, number of bits = 13117.41, Y-PSNR = 42.09 dB, time = 174.22 sec, QP = 22.

TABLE III
SUMMARY OF ALL ENCODING RESULTS EXPRESSED IN PERCENTAGE (%).

Cl.	Sequence	Resolution	Proposed Method			HIDR-CNN [11]		
			△ PSNR	**△ Bit Rate**	**△ Enc. Time**	**△ PSNR**	**△ Bit Rate**	**△ Enc. Time**
A	$Traffic$	2560 × 1600	−0.28	5.83	44.91	−0.04	0.75	30.60
B	$BQTerrace$	1920 × 1080	−0.19	3.74	45.00	− − −	− − −	− − −
	$Cactus$		−0.17	4.87	49.49	−0.10	2.01	33.95
	$ParkScene$		−0.21	4.97	44.49	−0.07	1.80	36.87
C	$BasketballDrill$	832 × 480	−0.18	3.97	48.17	0.00	0.08	15.42
	$BQMall$		−0.21	3.86	46.88	− − −	− − −	− − −
	$PartyScene$		−0.05	0.64	37.66	−0.03	0.43	19.71
	$RaceHorses$		−0.16	2.73	46.25	−0.02	0.26	15.56
D	$BasketballPass$	416 × 240	−0.31	5.52	46.38	0.00	0.04	18.98
	$BlowingBubbles$		−0.08	1.37	38.04	−0.01	0.43	13.80
	$BQSquare$		−0.04	0.47	45.21	− − −	− − −	− − −
	$RaceHorses$		−0.13	2.17	41.95	−0.02	0.27	9.58
Average			**−0.17**	**3.35**	**44.54**	**−0.03**	**0.67**	**21.61**

within the depth range, facilitating the selection of an optimal partitioning structure. However, it's important to note that this learning-based model necessitates online learning, introducing additional time overhead. Consequently, the proposed model outperforms the reference model in terms of encoding time efficiency.

2) **Analysis of RD Performance:** The Table III also provides a comparison of the rate-distortion (RD) performance with an emphasis placed on the increase in bit rate. Proposed model shows an average 3.35% increase in bit rate across all test video sequences while for the $BQSquare$ sequence increase is only 0.47%. However, the $Traffic$ sequence shows highest increase in bit rate, which is 5.83% among all the test video sequences. The results indicate that the overall bit-rate gain is considerable as our main focus is reducing time complexity.

V. CONCLUSION

In this research, we propose a novel method to simplify encoding process in HEVC's intra-mode, utilizing the zero-cross edge detection method. The proposed method focuses on finding one of the best CTU partition as opposed to the conventional RDO brute-force search. While it substantially reduces complexity and saves time, there is a modest increase in bit-rate. Our comprehensive analysis covering various video patterns, resolutions, and block sizes validates the method's effectiveness. The primary challenge was achieving quicker CTU partitioning decisions. The proposed method uses zero-cross edge information to minimize the unnecessary partitioning complexity and enhance encoding efficiency. Our model succeeds over the original HM solution in performance. The result shows 44.54% reduction in encoding time. Changes in the Peak Signal-to-Noise Ratio (PSNR) is very minor, while the bit-rate shows a considerable increase of 3.35%. This demonstrates our approach's success in advancing video coding technology.

VI. Acknowledgement

The authors express gratitude to the ICT division, Bangladesh, for furnishing partial funding that facilitated the completion of this research endeavor.

References

[1] I. E. Richardson, *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.

[2] M. E. H. Eimon, M. Z. Islam, M. S. U. Zaman, M. A. M. Hasan, and B. Ahmed, "Superpixel based inter-frame prediction for video coding," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 2019, pp. 409–413.

[3] 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.

[4] M. Z. Islam and B. Ahmed, "A machine learning approach for fast cu partitioning and time complexity reduction in video coding," in *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*, 2021, pp. 1–5.

[5] A. Heindel, T. Haubner, and A. Kaup, "Fast CU Split Decisions for HEVC Inter Coding Using Support Vector Machines," in *2016 Picture Coding Symposium (PCS)*. IEEE, 2016, pp. 1–5.

[6] M. Z. Islam and B. Ahmed, "Visualization of ctu partitioning and time complexity analysis in hevc and vvc," in *2021 3rd International Conference on Electrical Electronic Engineering (ICEEE)*, 2021, pp. 169–172.

[7] I. Werda, A. Maraoui, F. E. Sayadi, and N. Masmoudi, "Fast cu partition and intra mode prediction method for hevc," in *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2022, pp. 562–566.

[8] N. Westland, A. S. Dias, and M. Mrak, "Decision Trees for Complexity Reduction in Video Compression," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2666–2670.

[9] X. Chen, Y. Zhang, Y. Li, and J. Wen, "Decision Tree Based Inter Partition Termination For Av1 Encoding," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1585–1589.

[10] M. Z. Islam, M. E. H. Eimon, B. Ahmed, and M. A. M. Hasan, "Classification based inter-frame prediction in video compression," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 2019, pp. 404–408.

[11] Z. Feng, P. Liu, K. Jia, and K. Duan, "Hevc fast intra coding based ctu depth range prediction," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 2018, pp. 551–555.

[12] A. Feng, C. Gao, L. Li, D. Liu, and F. Wu, "Cnn-based depth map prediction for fast block partitioning in hevc intra coding," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[13] P. Hari, V. Jadhav, and B. S. Rao, "Ctu partition for intra-mode hevc using convolutional neural network," in *2022 IEEE International Symposium on Smart Electronic Systems (iSES)*, 2022, pp. 548–551.

[14] X. Hou and Y. Xue, "Fast coding unit partitioning algorithm for hevc," in *2014 IEEE International Conference on Consumer Electronics (ICCE)*, 2014, pp. 7–10.

[15] N. Kim, S. Jeon, H. J. Shim, B. Jeon, S.-C. Lim, and H. Ko, "Adaptive keypoint-based cu depth decision for hevc intra coding," in *2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2016, pp. 1–3.

[16] M. Jamali, S. Coulombe, and F. Caron, "Fast hevc intra mode decision based on edge detection and satd costs classification," in *2015 Data Compression Conference*, 2015, pp. 43–52.

[17] B. Chung and C. Yim, "Fast intra prediction method by adaptive number of candidate modes for rdo in hevc," *Information Processing Letters*, vol. 131, pp. 20–25, 2018.

[18] A. BenHajyoussef, T. Ezzedine, and A. Bouallègue, "Gradient-based pre-processing for intra prediction in high efficiency video coding," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 9, Jan 2017.

[19] Y. Huang, L. Song, R. Xie, E. Izquierdo, and W. Zhang, "Modeling acceleration properties for flexible intra hevc complexity control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4454–4469, 2021.

[20] B. Huang, Z. Chen, K. Su, J. Chen, and N. Ling, "Low-complexity rate-distortion optimization for hevc encoders," *IEEE Transactions on Broadcasting*, vol. 67, no. 3, pp. 721–735, 2021.

[21] V. V. Menon, H. Amirpour, C. Timmerer, and M. Ghanbari, "Incept: Intra cu depth prediction for hevc," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–6.

[22] E. Çetinkaya, H. Amirpour, M. Ghanbari, and C. Timmerer, "Ctu depth decision algorithms for hevc: A survey," *Signal Processing: Image Communication*, vol. 99, p. 116442, 2021.

[23] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the hevc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1697–1706, 2012.

[24] M. Z. Islam and B. Ahmed, "Fast cu splitting and computational complexity minimization technique by using machine learning algorithm in video compression," in *2021 3rd International Conference on Electrical Electronic Engineering (ICEEE)*, 2021, pp. 181–184.

[25] Y. Li, Y. Fang, P. Haipeng, and N. Ling, "Bagged tree and resnet-based joint end-to-end fast ctu partition decision algorithm for video intra coding," *Electronics*, vol. 11, p. 1264, 04 2022.

[26] R. C. Gonzalez and R. E. Woods, "Digital image processing," pp. 176–178, 2018.

[27] F. Bossen *et al.*, "Common Test Conditions and Software Reference Configurations," *JCTVC-L1100*, vol. 12, 2013.

[28] "HM Reference Software for High Efficiency Visdeo Coding (HEVC)," [Online], Available: https://hevc.hhi.fraunhofer.de/.

[29] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD-curves," *VCEG-M33*, 2001.