Received: September 16, 2023. Revised: October 27, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Coding Tree Unit Partition Prediction Algorithm Utilizing Residual Attention-Based Long Short-Term Memory for Video Intra Coding

Suhas Shankarnahalli Krishnegowda[1]* **Hosanna Princye Periapandi[1]**

[1]*Department of Electronics and Communication Engineering, S.E.A. College of Engineering and Technology, Bangalore, Visvesvaraya Technological University, Belagavi-590018, India*
* Corresponding author's Email: suhashoodi@gmail.com

Abstract: High-efficiency video coding (HEVC) stands as one of the most extensively employed video coding standards, with the primary encoding challenge revolving around the searching process. To minimize coding complexity of HEVC block partition, a new algorithm is proposed for supporting the partition architecture of coding tree unit (CTU) in intra-coding. In this research, problem of partition architecture decisions of CTU is resolved through two stages. In first phase, the bagged tree method is evaluated for predicting CTU splitting, and in the second phase, the partition issue of 32 × 32 size CU is given as 17 outcome classification. To attain a huge accuracy of prediction, a residual attention-based long short-term memory (LSTM) with parametric rectified linear unit (AT-LSTM-PReLU) is proposed. The proposed method produces a structure of partition quad-tree of CTU that makes multiple decisions at variant depth levels. The performance of proposed method is estimated with a dataset gathered from the joint collaborative team and video coding (JCT-VC). The proposed approach achieved a remarkable bit rate improvement of 3.72%, a significant ΔT enhancement of 97.28%, and minimized video quality degradation by just -0.05dB. These results outperform other deep learning methods like convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM).

Keywords: Bagged tree, Coding tree unit, High-efficiency video coding, Long short-term memory, Parametric rectified linear unit.

1. Introduction

The significant advancement in communication and multimedia technology has led to the emergence of virtual reality (VR), ultra-high definition (UHD), and high dynamic range (HDR) videos. These innovations have greatly enhanced the visual experience of high-efficiency video coding (HEVC) [1]. This increasing introduced of video information leads to issues on storage and transmission [2]. For transmitting a huge quantity of video information effectively at a better bit rate, the standards of video transmission are introduced like multiple function video coding standard (VVC/H.266), advanced video coding (AVC/H.264), and high-efficiency video coding (HEVC/H.265) [3]. The video coding standards mentioned in the text employ flexible block coding techniques and incorporate effective coding tools, thereby maximizing complexity of different video coding standards [4]. By considering VVC and HEVC, the HEVC takes a quad-tree-based CU recursive partition technique for CU [5]. The VVC utilizes quad-tree architecture as well as bi and trinomial tree partition architectures for better CU partition that made the complexity of intra-frame coding averagely 18 times greater than HEVC [6, 7]. Utilizing deep learning-based acceleration methods, both encoders demanding support from graphical processing units (GPUs) and diverse hardware GPUs aiding HEVC and VVC are enhanced [8].

The HEVC is much widely utilized than the VVC in the application of industries. The HEVC and VVC techniques utilize block-based encoding architectures along with variations in particular techniques [9]. HEVC incorporates numerous techniques originally employed in AVC/H.264 to significantly enhance video compression efficiency [10].

The quadtree of HEVC partitioning utilizes the search of brute force to Rate Distortion Optimization (RDO) cost measurement [11, 12]. The procedure complexity is high when utilized along normal signal processing stages which makes HEVC complex. The RDO process comprises two main steps: a comparison phase and a checking procedure. Initially, it evaluates the rate-distortion cost of the parent CTU and the total cost associated with splitting it until the end [13, 14]. In the comparison phase, it assesses the RD cost of the parent CTU and the cost after splitting. If the RD cost after splitting exceeds a predetermined threshold, further splitting is not feasible. Conversely, if the parent's RD cost is higher, it is permissible to proceed with the split operation [15]. This calculation process in HEVC is difficult which makes the model complex. In HEVC, the searching process of superior performance is the major complexity in encoding. A novel algorithm has been introduced to streamline the coding complexity of block partitioning in HEVC. This algorithm is designed to facilitate the determination of partition structures for CTU partitioning in intra-coding.

The major contribution of research is as follows:

- A two-phase prediction model with a mixture of bagged tree and residual attention-based LSTM is proposed which attains high accuracy prediction. The proposed method minimizes CU partition classes effectively and makes multiple decisions at variant depth levels.
- The residual attention-based LSTM with Parametric ReLU is proposed efficiently that skips unwanted search for various combinations of CU sizes. The PReLU activation function gives assurances for both non-linearity and overcomes the issue of neurons dying.
- The performance of proposed method is estimated by utilizing parameters of Bitrate, video quality degradation and time saving.

The remaining of the research paper is given as follows: Section 2 defines literature review and section 3 defines process of proposed methodology. Section 4 explains results of proposed method and comparative analysis. The conclusion of this research is given in section 5.

2. Literature review

Joy [16] implemented a deep convolutional neural network (DCNN) for the reduction of complexity in high efficiency video coding (HEVC). The implemented method predicted coding tree unit (CTU) through intra prediction process low complex HEVC. The model's complexity was estimated by incorporating it into an HEVC pipeline and using actual data. The implemented method reduces the encoding time and enhanced the model accuracy. The performance rate-distortion degradation was much penalizing.

Amna [17] introduced a deep convolutional neural network (DCNN) for reducing high efficiency video coding (HEVC) complexity in intra coding unit partition. The introduced method utilized two phases such as fuzzy support vector machine (FSVM) and CNN. The FSVM technique explored the result suggestions on the efficiency of coding and difficulty of encoding and the CNN method was utilized for partitioning the HEVC intra prediction. The introduced method reduces the complexity of HEVC by optimizing whole intra configurations however, the depth produces some complexity.

Imen [18] suggested modified LeNet-5 and AlexNet-based techniques for deleting an extensive execution time in high-efficiency video coding (HEVC). The CNN method was utilized for substituting rate-distortion optimization (RDO) search without damaging the performance of compression effectiveness. The LeNet-5 and AlexNet techniques were implemented for deleting extensive execution time, which optimized the partition module of the HEVC coding unit for the whole intra-configuration. The suggested method quickens the structure of the CU partition by minimizing intra-mode encoding time. The method cannot directly be applied to HEVC to decide the CU splitting decision.

Amna [19] developed a lightweight neural network (LNN) for the early direction determination strategy to ignore unnecessary multi-type tree (MTT) as well as minimize execution complexity. The developed method substitutes the rate-distortion optimization (RDO) search utilized without damaging the performance of compression effectiveness. Eventually, the developed model was utilized to predict the respective versatile video coding (VVC) TT partition that optimized VVC coding unit partition modules. The method did not explore the coding unit correlation partition among the frames.

Sairam and Muralidhar [20] presented a horizontal subsampling motion estimation (HSME) technique for identifying correct motion vectors with minimized complexity in high-efficiency video coding (HEVC). The presented method was utilized to accelerate the motion evaluation process and the

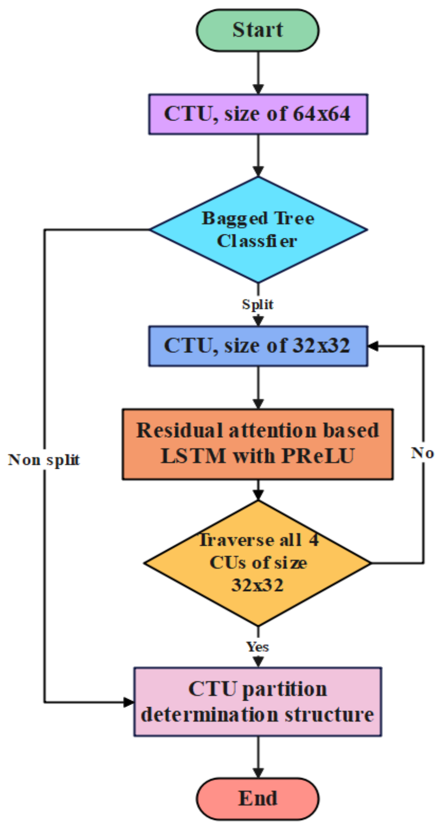

Figure. 1 Flowchart of the proposed method

ET-LSTM method was structured in the mixture through ET-CNN for predicting CTU partition architecture in HEVC. The presented technique quickens the motion evaluation process and acquires a motion vector with a global minimum. The method has huge execution complexity and does not produce much accurate prediction.

Liu [21] introduced multiple task intra-mode decision network (MID-Net) method for efficiently predicting the suitable angular modes in versatile video coding (VVC). The introduced method involved rough mode decision (RMD) and candidate mode list (CML). The probability of learning- and statistics-oriented are incorporated for enhancing the accuracy of prediction and making sure that unwanted intra-modes are skipped. The introduced method automatically extracted the suitable features and obtained superior coding performance. The method utilized a brute force technique to identify the minimum Rate Distortion (RD) cost.

Faragallah [22] implemented the Selective Encryption (SE) in chaotic video stream cipher to scalable high-efficiency video coding (SHVC) named SHVC-SE. Implemented method utilized a logistic map (LM) to encipher motion vector difference (MVD), coefficients of discrete cosine transform (DCT), and sign bits of delta quantization parameter (DQP). Implemented method secured many sensitive bits of SHVC through kept features of video format consent, but the method was more time consuming.

Galiano [23] developed Hybrid HEVC techniques such as Deep Learning and Parallel Computing to efficiently minimize the complexity of the HEVC encoding system. At first, the approach employed a parallel strategy that leveraged a domain decomposition method relying on HEVC partitioning. This approach was designed to make efficient use of the shared memory across multiple core processors. The further process utilized optimization techniques at the CTU stage to minimize the difficulty of the quad-tree separating process through CNN. The developed method has huge execution complexity and does not produce much accurate prediction.

The existing methods has limitation such as AlexNet they cannot be directly applied to HEVC to decide CU splitting decisions. The LNN method failed to explore the Coding Unit correlation partition among the frames. The Hybrid HEVC method has huge execution complexity and does not produce much accurate prediction. The performance of rate-distortion degradation was much penalizing in DCNN.

3. Proposed methodology

The quick coding tree unit (CTU) partition algorithm for high-efficiency video coding (HEVC) in intra-coding is proposed. The proposed model has two stages, The bagged tree model is utilized to first stage. Efficient features are determined for training bagged tree models. In the second stage, the attention-based LSTM with Parametric ReLU is developed to attain a better performance of prediction. The flowchart of proposed method is represented in Fig. 1.

In first phase, bagged tree classifier is utilized for predicting splitting of coding tree unit and either output is split or non-split. Particularly, when it is anticipated that the CTU is a non-split class, it proceeds directly to the process of determining the coding tree unit partition structure, as illustrated in Fig. 1. The ultimate prediction is the entirety of a 64×64 CTU without any splitting. Or else CTU is predicted for splitting through bagged tree model, it splits into 32×32 size of 4 sub-CUs. These 4 CUs can be passed through second phase for next processing. In second phase, 32×32 sized CU is given into a trained attention-based LSTM classifier. As an outcome of 17 labels, every of that shows partition architecture for CU with 32×32 size, a single label is the outcome of attention-based LSTM. Likewise, every 4 sub-CUs are predicted to split in

Figure. 2 Sample Frames of videos in the dataset

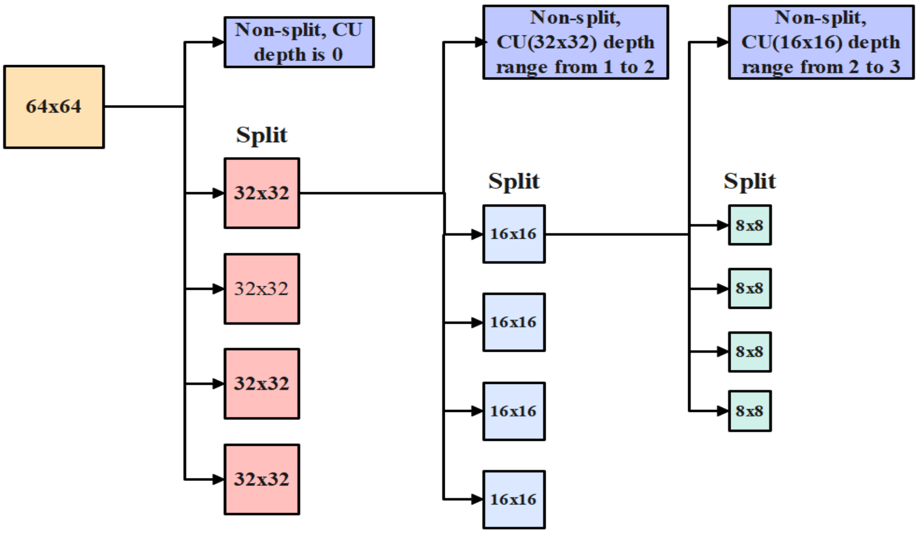

Figure. 3 Process of partition with quad tree

initial phase are classified and 4 simultaneous predicted labels are produced. Next, the 4 labels and parent CTUs are processed further.

3.1 Dataset

The dataset utilized in the research is gathered from joint collaborative team and video coding (JCT-VC) [24] for HEVC CU partitioning in every intra-configuration. The collected datasets have a total of 34 videos which are divided into 14 videos utilized for training, 6 videos utilized for validation, and 14 videos utilized for testing. The training and testing videos are selected randomly from different resolutions to ensure network efficiency and assess performance. By utilizing HEVC in the HM 16.20 model, each intra-configuration encodes whole video sequences for various QP values 22, 27, 32, and 37. Sample frames of videos in the dataset such as racehorse, vidyo1, BQ Square, party scene, basketball drive, and traffic are represented in Fig. 2.

3.2 Partition

For encoding video sequences, every frame between a sequence is separated into many squares without overlapping of size 64×64. The highest Coding Unit (CU) of size 64×64 is defined Coding Tree Unit (CTU). If the CU size is larger, it can accommodate a significant amount of semantic data, while a smaller CU can attain much precise pixel prediction values. The HEVC evaluates the process of recursive splitting of Coding Tree Unit and it is stopped when it reaches a small CU size. There are four various sizes of CUs such as 64×64, 32×32, 16×16 and 8×8 which are shown as various colors in Fig. 3.

3.3 Bagging

The bagged tree method is implemented as ensemble method for various decision tree models, that is trained through a randomly selected subset of training data. After training process, final prediction of bagged tree method for occurrence is done through considering major votes of prediction from each decision tress of input. The bagging model minimizes model variance without maximizing bias and bagged tree model performance is superior to individual decision tree model. So, in this research, bagged tree is utilized in first stage for predicting splitting outcome of the Coding Tree Unit. Fig. 4 represents the architecture of the bagged tree model, which has n decision trees, and how it produces final prediction.

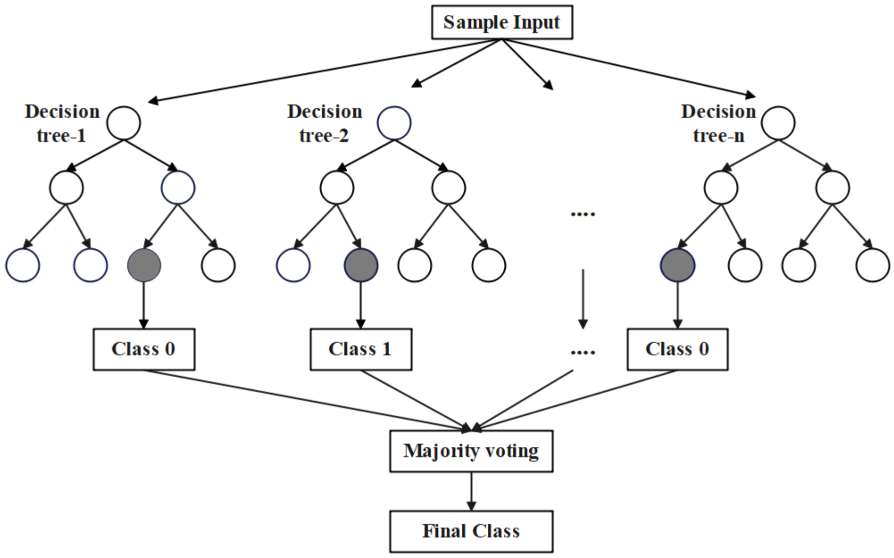

Figure. 4 Process of bagged tree model

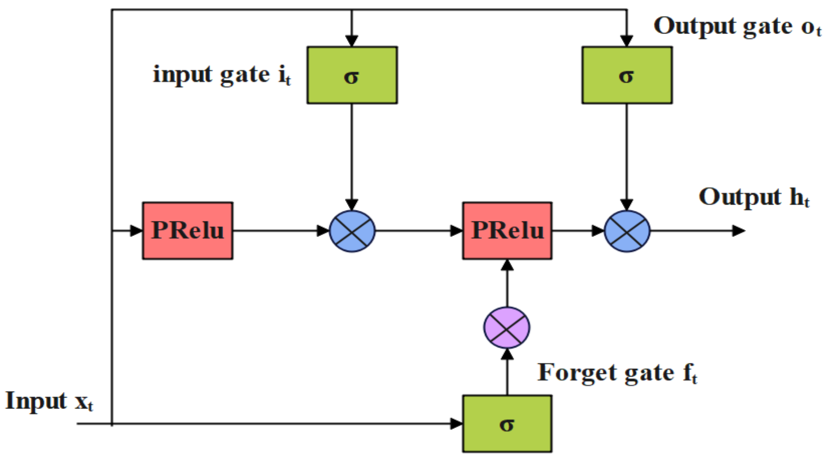

Figure. 5 Structure of input, output and forget gates

3.4 Long-short term memory

The LSTM is a kind of recurrent neural network (RNN) that learns long-term dependency, particularly in sequence prediction problems. It has huge memory power for remembering the outcomes of every node for a much extended time to generate the output for the next node effectively. Let us consider that x_t, h_t and C_t are input, control and cell state at time t. The $(x_1, x_2, \dots, x_m)$ are the sequence of inputs, LSTM evaluates h-sequence $(h_1, h_2, \dots, h_m)$ and C-sequence $(C_1, C_2, \dots, C_m)$. The mathematical formulae are represented in Eqs. (1-6) and Fig. 5 represents the structure of various gates.

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o \quad (3)$$

$$\bar{C}_t = PReLU(W_c . [h_{t-1}, x_t] + b_c) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (5)$$

$$h_t = o_t * PReLU(C_t) \quad (6)$$

Here, σ represents the sigmoid function,
$*$ represents element-wise multiplication,
W_f, b_f , W_i, b_i and W_o, b_o represents weight and bias of forget, input and output gate respectively,
W_C, b_C represents weight and bias of cell state,
i_t, o_t *and* f_t represents input, output and forget gates respectively. Every LSTM unit contains a memory cell state C_t at time t, which is managed by these three gates.

3.4.1. Attention based LSTM

Attention based neural networks have attained numerous techniques in the prediction and classification of images/videos. The different attention mechanisms are developed and the residual based attention model is utilized in the research for obtaining an active weighted sum of features extracted from the frames of video. Rather than using a frame as LSTM input, utilized characteristics of weighted image which compensates the attention in residual attention model. Residual based attention is employed to specify high-level layers, sequence data, and layer formulation by searching residual functions that are given to the input layer. The mathematical formula of residual learning function is represented as Eq. (7),

$$Y = f\left(\ddot{X}, \ddot{W}\right) + \ddot{X} \tag{7}$$

Where, $\ddot{X}$ and Y represents the input and output sequential data vectors of layers.

$f\left(\ddot{X}, \ddot{W}\right)$ represents the residuals learned from relative layer.

The outcomes of those layers in residual learning create sequence of fed input and nonlinear residual. The major benefit of the method is it develops shortcut functions between certain layers for highly efficient model training as well as beneficial for protecting the vanishing gradients problems. This research normalizes the data by evaluating normalization in residual LSTM to an easy effective hidden state, normalizes data of neurons in LSTM as well and minimizes the training time. The mathematical formula is represented by Eqs. (8-10),

$$\tilde{n}_t = \frac{1}{h}\sum_{i=0}^{h}(ht)i \tag{8}$$

$$\delta_t = \sqrt{\frac{1}{h}\sum_{i=0}^{h}((ht)i - \tilde{n}_t)^2} \tag{9}$$

$$\acute{y}_t = f\left(\frac{\dot{g}}{\delta_t} \odot (ht - \tilde{n}_t) + b\right) \tag{10}$$

Here, $(ht)i$ represents hidden state in every LSTM layer of ith neuron,

$\dot{g}$ and b represents trainable weights which are utilized for reducing input sequence of activation function f,

t is represented as a time step,

The threshold value is applied to 0.5 in every residual LSTM layer to minimize overfitting. The research utilized compression and decompression with attention mechanism-based LSTM to improve the performance of video encoding. It assessed the decompression process used in block generation, where video features are fed into the next block, relying on frames previously generated by model. This technique efficiently produces video frames utilizing two kinds of inputs such as natural language processing $1D$ feature vector and $2D$ video frame information. The research utilized a PreLU activation layer with residual LSTM which evaluates the short- and long-term dependency using latent correlation between features in different locations. In the context of video encoding models, the input consists of video frames provided to a residual attention-based LSTM. This LSTM necessitates a feature block for sequence learning. The mathematical formulae are represented from Eqs. (11-13),

$$k_t = \frac{1}{h}\sum_{i=1}^{h}\widehat{W}_t^i f_i \tag{11}$$

$$S_t = \widehat{W}^T PReLU(\widehat{W}_h ht + M_h R_f + b_h \tag{12}$$

$$A_t = PReLU\ (S_t) \tag{13}$$

Where, $\widehat{W}^T$, $\widehat{W}_h$, M_h and b_h represents parameters learned for features of the frame f_i respect to attention to weight $\widehat{W}_t^i$ for returning to the score S_t. A_t represents the output. The extracted 35-frame sequences are passed through residual attention-based LSTM and the final prediction is done by utilizing the Parametric ReLU layer.

3.4.2. Parametric ReLU

The major disadvantage of ReLU is the $x < 0$ part in the ReLU activation function which can cause the deaths of neurons. To overcome the issue of neurons dying, the part function needs to generate gradients and this part requires to be transformed. The parametric rectified linear unit (PReLU) activation function is an attempt to overcome the issue of dying neurons in ReLU. The PreLU produces a relu function with negative slope α, when $x \geq 0$, function is not 0, but a less negative slope, where α is the learnable parameter. If α is stable, it is known as Leaky ReLU. The Parametric ReLU activation function merges quicker and contains less training error as well and the implementation of α parameter in activation function cannot lead to overfitting. In this research, the Residual attention-based LSTM with Parametric ReLU activation function is utilized for video coding CU partitioning that saves time and bitrate with less video quality degradation. The mathematical formula of Parametric Rectified Linear Unit is represented in Eq. (14),

$$PReLU(x) = \begin{cases} x & if\ x \geq 0 \\ \alpha x & if\ x < 0 \end{cases} \tag{14}$$

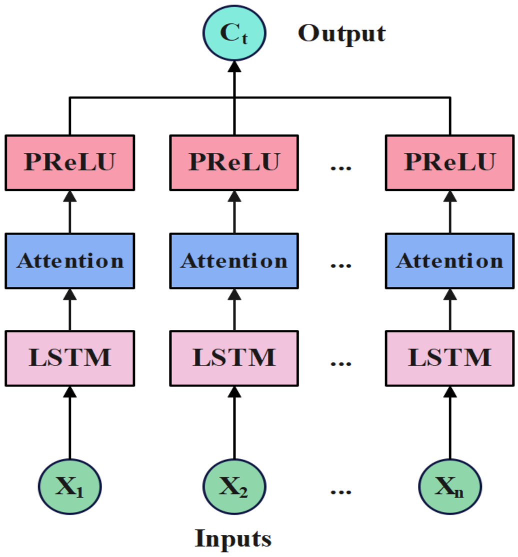

Figure. 6 The process of proposed Residual attention-based LSTM with PReLU

The α parameter is not equal to 1, this PReLU activation function gives assurances for both non-linearity and overcomes the issue of neurons dying. Fig. 6 represents the process of the proposed residual attention-based LSTM with PReLU.

The partition of CTU architecture determination process is evaluated by above mentioned two phases. Following 4 labels produced by attention-based LSTM in the second phase, partition architecture of respective parent CTU is determined. Predicted label (PL) results through attention-based LSTM for CU 32×32 size from $1\ to\ 17$. The respective partition quad tree architecture is represented in Fig. 7 here, values of PL for 4 sub-CUs (as 1, 2, 3, and 4) are considered as 1, 5, 17 and 2.

Partition of CTU is produced, optimum rate-distortion cost is measured directly without huge comparisons. Next, the process of encoding is performed. When compared with classical recursive RDO, the proposed residual attention-based LSTM algorithm efficiently skips unwanted searches for various mixtures of CU sizes. Also, compared with various previous quick partition models introduced with depth level, this research proposed an efficient two-phase CTU partition algorithm utilizing bagged tree and residual attention-based LSTM techniques. The output of the CTU partition is measured through the bagged tree model and residual attention-based LSTM model. The proposed method efficiently minimizes time utilization on RDO and spends minimum time on training.

4. Performance analysis

In this research, the proposed residual attention-based LSTM with Parametric ReLU model is simulated by MATLAB R2020a software. The simulation is performed with an i7 processor system with 8 GB random access memory and 1 TB hard disk. The performance of proposed method is estimated by utilizing parameters such as Bit Rate (ΔBR), Video quality degradation (ΔPSNR) and encoding time-saving (ΔT). The mathematical formulae of the parameters are represented in Eqs. (15-17)

$$\Delta BR = \frac{1}{4}\sum_{i=1}^{4} \frac{BR_{prop}(QP_i) - BR_{org}(QP_i)}{BR_{prop}(QP_i)} \times 100 \quad (15)$$

$$\Delta PSNR = \frac{1}{4}\sum_{i=1}^{4} PSNR_{prop}(QP_i) - PSNR_{org}(QP_i) \times 100 \quad (16)$$

$$\Delta T = \frac{1}{4}\sum_{i=1}^{4} \frac{T_{prop}(QP_i) - T_{org}(QP_i)}{T_{prop}(QP_i)} \times 100 \quad (17)$$

4.1 Quantitative analysis

The performance of proposed residual attention-based LSTM is evaluated by utilizing the parameters of ΔBR, ΔPSNR and ΔT. In Table 1, the effectiveness of the utilized activation function is evaluated with various activation functions like sigmoid, softmax, and rectified linear unit (ReLU). The utilized Parametric ReLU attained a high bitrate of 1.72% and saved more encoding time of 87.44% with less degradation of video quality of -0.15dB which is comparatively superior to other activation functions such as sigmoid function saves encoding time of 70.84%, softmax function saves encoding time of 75.68% and ReLU saves time of 82.57%.

In Table 2, the effectiveness of the proposed residual attention-based LSTM is evaluated with various deep learning techniques like convolutional neural network (CNN), recurrent neural network (RNN) and long short-term memory (LSTM). The proposed Residual attention-based LSTM attained a high bit rate of 3.72% and saved more encoding time at 97.28% with less degradation of the video quality of -0.05dB which is comparatively better than other deep learning techniques such as CNN saves time of 75.06%, RNN saves encoding time of 77.28% and LSTM saves time of 88.73%.

In Table 3, the effectiveness of the proposed method is evaluated with various sequences of videos such as traffic, basketball drive, BQ mall, party scene, BQ square, race horses, and Vidyol. The average of these sequences is taken into consideration of

Received: September 16, 2023. Revised: October 27, 2023.

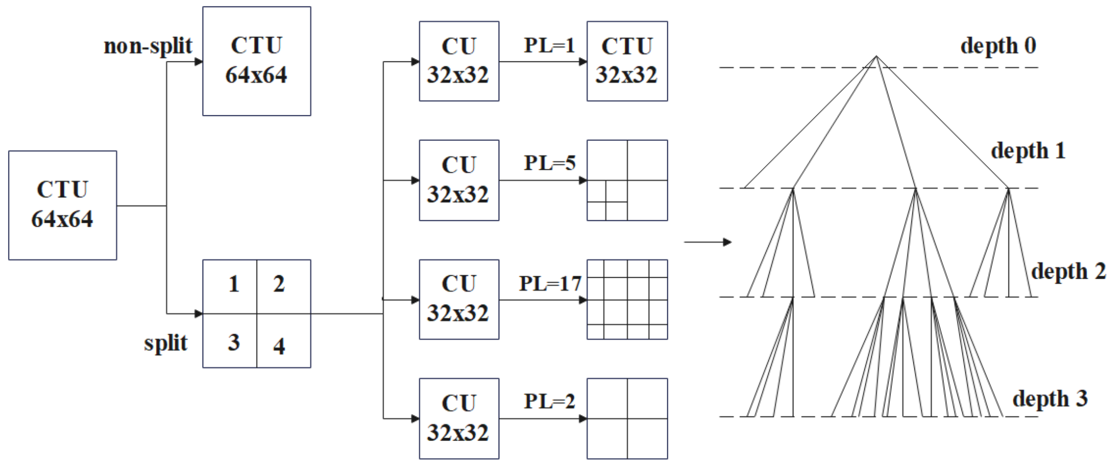

Figure. 7 CTU partition architecture determination process

Table 1. Performance of activation function

Activation functions	ΔBR (%)	ΔPSNR (dB)	ΔT (%)
Sigmoid	0.48	-0.27	70.84
Softmax	0.69	-0.19	75.68
ReLU	1.02	-0.22	82.57
Parametric ReLU	1.72	-0.15	87.44

Table 2. Performance of proposed method with various deep learning methods

Methods	ΔBR (%)	ΔPSNR (dB)	ΔT (%)
CNN	1.44	-0.24	75.06
RNN	1.50	-0.16	77.28
LSTM	2.52	-0.21	88.73
Residual attention-based LSTM	3.72	-0.05	97.28

Table 3. Performance of proposed method with various sequences of videos

Residual attention-based LSTM with Parametric ReLU			
Sequences	ΔBR (%)	ΔPSNR (dB)	ΔT (%)
Traffic	3.04	-0.07	95.23
Basket Ball Drive	4.13	-0.10	98.31
BQ Mall	3.45	-0.02	98.03
Party Scene	3.93	-0.04	97.21
BQ Square	3.86	-0.05	96.72
Race Horses	3.67	-0.03	98.09
Vidyol	4.01	-0.06	97.42
Average	3.72	-0.052	97.28

parameters ΔBR, ΔPSNR and ΔT. The ΔBR of various sequences are traffic attained 3.04, basketball drive attained 4.13, BQ mall attained 3.45, party scene attained 3.93, BQ square attained 3.86, race squares attained 3.67, Vidyol attained 4.01 and the average of ΔBR of these various sequences is 3.72. The ΔPSNR of various sequences are traffic attained -0.07, basketball drive attained -0.10, BQ mall attained -0.02, party scene attained -0.04, BQ square attained -0.05, race squares attained -0.03, Vidyol attained -0.06 and the average of ΔPSNR is -0.052. The ΔT of various sequences are traffic attained 95.23, basketball drive attained 98.31, BQ mall attained 98.03, party scene attained 97.21, BQ square attained 96.72, race squares attained 98.09, Vidyol attained 97.42 and the average of ΔT of these various sequences is 97.28. Fig. 8 represents the compressed frames of certain sample frames. In Table 4, comparison of encoding time is described.

4.2 Comparison analysis

In this section, the comparative analysis of proposed residual attention-based LSTM with PReLU method is described in Table 5. The performance of proposed method is compared with other existing methods such as lightweight neural network (LNN) [19], horizontal subsampling motion estimation (HSME) [20] and hybrid HEVC [23] under JCT-VC dataset. The proposed method attained a high bit rate of 3.72% and saved encoding time of 97.28% which is comparatively higher than other existing methods LNN attained a bit rate of 0.74% and saved encoding time of 46.91%, the HSME attained a bit rate of 1.68% and saves time of 59% and hybrid HEVC attained bit rate of 3.22% and saves encoding time 0f 96.10%.

4.3 Discussion

The existing research such as LeNet-5 and AlexNet [18] has limitations in that they cannot be directly applied to HEVC to decide CU splitting

Table 4. Encoding time comparison

Sequences (Resolution size)	Encoding time with HEVC in Residual attention-based LSTM with PReLU (T1)	Encoding time with original HEVC (T2)	T1 – T2
Traffic (2560 x 1600)	187.194	474.175	-286.981
Basket Ball Drive (1920 x 1080)	123.372	537.640	-414.268
BQ Mall (832 x 480)	89.321	228.531	-139.21
Party Scene (1920 x 1080)	123.372	537.640	-414.268
BQ Square (416 x 240)	72.184	153.245	-81.061
Race Horses (832 x 480)	89.321	228.531	-139.21
Vidyol (416 x 240)	72.184	153.245	-81.061

Figure. 8 Compressed images of sample frames

Table 5. Comparative analysis

Author	Dataset	Method	ΔBR (%)	ΔT (%)
Amna [19]	JCT-VC	LNN	0.74	46.91
Sairam and Muralidhar [20]		HSME	1.68	59
Galiano [23]		Hybrid HEVC	3.22	96.10
Proposed method		Residual attention-based LSTM with PReLU	3.72	97.28

decisions. The LNN [19] method failed to explore the Coding Unit correlation partition among the frames. The hybrid HEVC [20] method has huge execution complexity and does not produce much accurate prediction. In HEVC, the searching process of superior performance is the major complexity in encoding. To minimize the coding complexity of HEVC block partition, a new algorithm is proposed for supporting the partition architecture of coding tree unit (CTU) in intra-coding. To attain a huge accuracy of prediction, a residual attention-based LSTM with parametric rectified linear unit (AT-LSTM-PReLU) is proposed. The proposed method minimizes CU partition classes effectively and makes multiple decisions at variant depth levels. The residual attention-based LSTM with Parametric ReLU is proposed to skip unwanted searches for various combinations of CU sizes. The PReLU activation function gives assurances for both non-linearity and overcomes the issue of neurons dying. The proposed method attained a high bit rate of 3.72% and saved more encoding time at 97.28% with less degradation of video quality of -0.05dB which is comparatively better than other deep learning techniques.

5. Conclusion

In HEVC, the searching process has the major complexity in encoding. To minimize the coding complexity of block partition in HEVC, a novel algorithm is proposed for supporting partition architecture decisions of coding tree unit (CTU) in intra-coding. In this research, problem of the partition architecture of CTU is resolved in two stages. In first phase, bagged tree method is evaluated for predicting CTU splitting and in the second phase, the partition issue of 32×32 size CU is given as 17 outcome classification, so it is resolved by one prediction. To attain a huge accuracy of prediction, a residual attention-based LSTM with parametric rectified linear unit (AT-LSTM-PReLU). The performance of proposed method is estimated with dataset gathered from joint collaborative team and video coding (JCT-VC). The proposed method attained a high bit rate of 3.72%, ΔT of 97.28%, and less video quality degradation of -0.05dB which is comparatively better than other deep learning methods like CNN, RNN and LSTM. The proposed method efficiently minimizes time utilization on RDO and spends minimum time on training. In the future, the transfer learning methods will be taken into consideration to enhance the generalization ability of LSTM.

Notation

Notations	Description
σ	Sigmoid function
$*$	Element-wise multiplication
W_f	Weight of forget gate
b_f	Bias of forget gate
W_i	Weight of input gate
b_i	Bias of input gate
W_o	Weight of output gate
b_o	Bias of output gate
W_C	Weight of cell state
b_C	Bias of cell state
i_t	Input gate
o_t	Output gate
f_t	Forget gate
C_t	Memory cell state
$\ddot{X}$	Input of sequential data
Y	Output of sequential data
$f(\ddot{X}, \ddot{W})$	Residuals learned from relative layers.
$(ht)i$	Hidden state
$\dot{g}$ and b	Trainable weights
t	Time step
$\widehat{W}^T$, $\widehat{W}_h$, M_h and b_h	Parameters learned for features
$\widehat{W}_t^i$	Weight attention
S_t	Score
A_t	Output
α	Parameter
ΔBR	Bit Rate
ΔPSNR	Video quality degradation
ΔT	Time-saving

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

For this research work all authors' have equally contributed in Conceptualization, methodology, validation, resources, writing—original draft preparation, writing—review and editing.

References

[1] J. Zhao, P. Li, and Q. Zhang, "A Fast Decision Algorithm for VVC Intra-Coding Based on Texture Feature and Machine Learning", *Computational Intelligence and Neuroscience*, Vol. 2022, p. 7675749, 2022.

[2] Z. Peng and L. Shen, "A classification-prediction joint framework to accelerate QTMT-based CU partition of inter-mode VVC", *Electronics Letters*, Vol. 59, No. 7, p. e12770, 2023.

[3] I. Taabane, D. Menard, A. Mansouri, and A. Ahaitouf, "Machine learning based fast QTMTT partitioning strategy for VVenC encoder in intra coding", *Electronics*, Vol. 12, No. 6, p. 1338, 2023.

[4] T. Wang, G. Wei, H. Li, T. Bui, Q. Zeng, and R. Wang, "A Method to Reduce the Intra-Frame Prediction Complexity of HEVC Based on D-CNN", *Electronics*, Vol. 12, No. 9, p. 2091, 2023.

[5] Y. Wang, Y. Liu, J. Zhao, and Q. Zhang, "Fast CU partitioning algorithm for VVC based on multi-stage framework and binary subnets", *IEEE Access*, Vol. 11, pp. 56812-56821, 2023.

[6] S. Wu, J. Shi, and Z. Chen, "HG-FCN: Hierarchical grid fully convolutional network for fast VVC intra coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 8, pp. 5638-5649, 2022.

[7] Z. Sun, L. Yu, and W. Peng, "QTMT-LNN: A fast intra CU partition using lightweight neural network for 360-degree video coding on VVC", *IET Image Processing*, Vol. 17, No. 2, pp. 597-612, 2023.

[8] J. T. Fang, B. H. Liu, and P. C. Chang, "Fast coding unit partitioning algorithms for versatile video coding intra coding", *Journal of Visual Communication and Image Representation*, Vol. 87, p. 103542, 2022.

[9] G. Li, J. Lei, Z. Pan, B. Peng, and N. Ling, "Multiple resolution prediction with deep up-sampling for depth video coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 9, pp. 6337-6346, 2022.

[10] J. Zhao, Y. Wang, M. Li, and Q. Zhang, "Fast coding unit size decision based on deep reinforcement learning for versatile video coding", *Multimedia Tools and Applications*, Vol. 81, No. 12, pp. 16371-16387, 2022.

[11] X. Shang, G. Li, X. Zhao, H. Han, and Y. Zuo, "Fast CU size decision algorithm for VVC intra coding", *Multimedia Tools and Applications*, Vol. 82, No. 18, pp. 28301-28322, 2023.

[12] Y. Song, B. Zeng, M. Wang, and Z. Deng, "An efficient low-complexity block partition scheme for VVC intra coding", *Journal of Real-Time Image Processing*, Vol. 19, No. 1, pp. 161-172, 2022.

[13] Y. Rahimi, M. Rezaei, and P. Jafari, "Low-complexity CNN-based CU partitioning for intra frames", *Journal of Real-Time Image Processing*, Vol. 20, No. 4, p. 73, 2023.

[14] S. B. Jdidia, F. Belghith, M. Jridi, and N. Masmoudi, "A multicriteria optimization of the discrete sine transform for versatile video coding standard", *Signal, Image and Video Processing*, Vol. 16, No. 2, pp. 329-337, 2022.

[15] K. Choi, T. V. Le, Y. Choi, and J. Y. Lee, "Low-complexity intra coding in versatile video coding", *IEEE Transactions on Consumer Electronics*, Vol. 68, No. 2, pp. 119-126, 2022.

[16] H. K. Joy, M. R. Kounte, and B. K. Sujatha, "Design and Implementation of deep depth decision algorithm for complexity reduction in High Efficiency Video Coding (HEVC)", *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 1, pp. 553-560, 2022.

[17] M. Amna, W. Imen, B. Soulef, and S. F. Ezahra, "Machine Learning-Based approaches to reduce HEVC intra coding unit partition decision complexity", *Multimedia Tools and Applications*, Vol. 81, No. 2, pp. 2777-2802, 2022.

[18] W. Imen, M. Amna, B. Fatma, S. F. Ezahra, and N. Masmoudi, "Fast HEVC intra-CU decision partition algorithm with modified LeNet-5 and AlexNet", *Signal, Image and Video Processing*, Vol. 16, No. 7, pp. 1811-1819, 2022.

[19] M. Amna, W. Imen, and S. F. Ezahra, "Fast multi-type tree partitioning for versatile video coding using machine learning", *Signal, Image and Video Processing*, Vol. 17, No. 1, pp. 67-74, 2023.

[20] S. K. Sairam and P. Muralidhar, "A Deep Learning Approach in Scalable High Efficiency Video Coding for Fast Coding Unit Size Decision", *IETE Technical Review*, Vol. 40, No. 3, pp. 287-302, 2023.

[21] Z. Liu, T. Li, Y. Chen, K. Wei, M. Xu, and H. Qi, "Deep Multi-task Learning based Fast Intra-mode Decision for Versatile Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[22] O. S. Faragallah, A. I. Sallam, M. Alajmi, and H. S. E. Sayed, "Low-latency selective encryption scheme for scalable extension of high-efficiency video coding", *Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 11, p. e4599, 2022.

[23] V. Galiano, H. Migallón, M. M. Rach, O. L. Granado, and M. P. Malumbres, "On the use of deep learning and parallelism techniques to significantly reduce the HEVC intra-coding time", *The Journal of Supercomputing*, Vol. 79, No. 11, pp. 11641-11659, 2023.

[24] Dataset: F. Bossen, "Common test conditions and software reference configurations", *9th Meeting of the JCT-VC*, JCTVC-L1100, Geneva, 2013.