

Received September 20, 2021, accepted October 22, 2021, date of publication November 8, 2021, date of current version November 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125697

# Fast Screen Content Coding in HEVC Using Machine Learning

EMAD BADRY<sup>1,2,3</sup>, (Member, IEEE), KOJI INOUE<sup>1,4,5</sup>, (Member, IEEE),  
AND MOHAMMED SHARAF SAYED<sup>2,6</sup>, (Member, IEEE)

<sup>1</sup>E-JUST Center, Kyushu University, Fukuoka 819-0382, Japan

<sup>2</sup>Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology, Alexandria 21934, Egypt

<sup>3</sup>Department of Electrical Engineering, Suez Canal University, Ismailia 41522, Egypt

<sup>4</sup>Department of Advanced Information Technology, Kyushu University, Fukuoka 819-0382, Japan

<sup>5</sup>System LSI Research Center, Kyushu University, Fukuoka 819-0382, Japan

<sup>6</sup>Department of Electronics and Communications Engineering, Zagazig University, Zagazig 44519, Egypt

Corresponding author: Emad Badry (emad.mahmoud@ejust.edu.eg)

This work was supported in part by the Egypt-Japan University of Science and Technology (E-JUST), in part by Kyushu University, and in part by the Egyptian Ministry of Higher Education (MoHe).

**ABSTRACT** Screen Content (SC) videos require proper tools to handle their special characteristics since they include repeated regions, sharp edges, and limited number of colors. Therefore, SC extension to High Efficiency Video Coding (HEVC) standard has been released for this purpose. SC extension has new tools such as Intra Block Copy (IBC) and Palette (PLT) mode. These tools improve the coding efficiency, but they come with a huge computation complexity. In this paper, we propose a scheme to reduce the encoding time of SC encoder. It has two algorithms based on the Decision Rule (DR) machine learning technique. The first algorithm, which is called Mode Skipping (MS), is used to skip the unnecessary SC mode checking. Early Pruning Termination (EPT), which is the second algorithm, is used to stop the partitioning process early. A small number of features has to be calculated for the trained models. The proposed scheme was implemented then simulated by using the standard software test model SCM-6. The experimental results show that the time complexity is reduced by 37.84% on average while the Bjøntegaard delta bit-rate (BD-R) increases by 1.34% only.

**INDEX TERMS** High efficiency video coding (HEVC), screen content (SC), decision rule.

## I. INTRODUCTION

Due to the popularity of screen sharing and computer graphics applications, Screen Content Coding (SCC) has become important. Therefore, the SCC extension to the High Efficiency Video Coding (HEVC) [1]–[3] has been released by the Joint Collaborative Team on Video Coding (JCT-VC), and that to enhance the coding of SC sequences over the conventional HEVC [4], which is suitable for Natural Content (NC) sequences.

In SCC, new tools have been incorporated over conventional Intra mode [5], such as Intra Block Copy (IBC) [6]–[8] and Palette (PLT) mode [9]–[11]. These tools utilize the special characteristics of the SC videos to improve the coding efficiency, since SC sequences contain repeated patterns, complex structure, and a limited number of colors. In IBC,

The associate editor coordinating the review of this manuscript and approving it for publication was Gulistan Raja.

a searching process is carried out to find the matched block of the current one in a similar manner to the Inter motion estimation [12] in conventional HEVC, but the search range is restricted to current frame. Although these additional tools improve the SC coding, the enhancement comes at the expense of a higher complexity.

In order to reduce the encoding complexity brought by using the additional SC tools, numerous approaches have been proposed. These efforts can be categorized into heuristic and machine learning-based works. In the case of heuristic approaches, the authors in [13] proposed to speed up the IBC mode according to the Rate-Distortion (RD) cost of Intra mode and the Coding Unit (CU) activity value. The RD cost was used in [14] to early terminate the mode testing process and the CU partitioning process. In [15], a hash-based scheme was suggested along the full frame. In which, the current CU finds the matched block among the CUs that have the same hash key. The authors in [16] speed up the local search of

IBC by using suggested hash keys for the different Prediction Units (PUs). In [17], zero activity and low gradient CUs are not checked in IBC mode. In [18], the temporal correlation between the current CU and its corresponding CU at previous frame is utilized to reduce the encoding time. Furthermore, an adaptive step size method was suggested to increase the searching speed of the IBC mode. In [19], the entropy and coding bits were used to achieve a fast CU decision. In [20], the CUs were classified into NC or SC in order to apply fast mode decision methods. Furthermore, the bit per pixel value is utilized to stop the splitting process early. A fast SC encoding scheme was proposed in [21] by using the pixel exactness value, the RD cost of the PLT mode, and suggested hash-based search method.

When it comes to the machine learning-based approaches, we proposed in [22] to use the Decision Tree (DT) technique to achieve fast mode decision. In addition, the luminance contrast inside the CU was used to perform fast CU size decision. Also, DT was used in [23] to boost up the SC encoding by inserting a decision block before the run of each mode. It introduced dynamic features and the intermediate RD costs for the training. In [24], the Random Forest (RF) was utilized to reduce the encoder complexity. A new hyperparameter tuning approach was introduced considering the encoding time and bitrate. The approaches in [25] and [26] used the Bayesian rule classification method. In [25], the authors applied a method for the corner point detection in order to categorize the frame into textual or pictorial regions. After that, the color number was employed for mode classification. In [26], an online-learning approach was proposed for the Bayesian rule to accelerate the mode decision and CU size decision. The modes of neighboring CUs were exploited to avoid the unnecessary mode checking. The authors in [27] proposed a scheme based on neural networks. The CUs were categorized as NC or SC type, and that to make fast mode decision. To achieve fast CU size determination, the correlation between the current CU and adjacent CUs was utilized. In [28], the complexity of intra prediction was reduced by using a Convolutional Neural Network (CNN). The global features were analyzed to give a decision if the mode would be checked or not. Then, local features guide the encoder to decide which modes should be tested.

In this paper, we present an approach for fast SC encoding. The reduction in encoding time is performed by utilizing SC modes skipping and early pruning termination for the CUs. We propose to use a small number of features for the training in order to alleviate the burden on the SC encoder. We select the Decision Rule (DR) machine learning technique because it can be easily implemented in SC encoders instead of using complex techniques such as in [28]. The proposed encoder does not have to extract features from neighboring CUs as in [24] or previous frame as in [18] which preserves the memory. In order to enhance the performance, we decide to apply the mode skipping for the IBC and PLT modes without skipping the conventional Intra mode.

The rest of the paper is organized as follows. The Intra mode decision of SC encoders is described in Section II. Section III explains the proposed decision rule based scheme. Section IV discusses the simulation results. Section V concludes the paper.

## II. INTRA MODE DECISION OF SC ENCODER

SC encoder utilizes the same partitioning method as the conventional HEVC encoder. The input frame is partitioned into square blocks. The main partitioning unit is called a Coding Tree Unit (CTU), which is  $64 \times 64$  pixels in size. The CTU is recursively divided into equally smaller four parts called CUs. Each CU can be  $2N \times 2N$ , where  $N$  can be 32, 16, 8, or 4. To achieve a better quality, the CU can be divided into smaller parts called Prediction Units (PUs), in addition, the PUs can be symmetric or asymmetric [4].

In order to determine the best structure of current CTU, the encoder compares the RD cost of current CU with the sum of the costs of its sub-CUs. Then, it terminates the partitioning process if the RD cost of current CU is the smaller. The optimum RD cost for each CU is determined after testing all prediction modes.

To find optimum RD cost in intra coding, the encoder starts to encode a CU with a mode called fast IBC, this mode is executed for the  $2N \times 2N$  CUs which have sizes less than  $64 \times 64$ . The block matching method is conducted in fast IBC through a number of Block Vectors (BVs). These BVs come from adjacent CUs and last checked CUs. After that, conventional Intra mode is examined for all CU sizes. This mode is skipped if zero distortion condition is satisfied for CUs that are tested by fast IBC. After finishing the test of Intra mode, the encoder runs IBC Skip/Merge mode, which is similar as the Skip/Merge mode in conventional HEVC. If Skip mode is the best one so far, the run of other SC tools is bypassed. Normal IBC is conducted for  $16 \times 16$  and  $8 \times 8$  CUs. In which, the encoder finds the best matched block for current  $2N \times 2N$   $16 \times 16$  PU along the whole frame in 1-D. When it comes to  $8 \times 8$  CUs, the best matched block is located by searching in a limited search region or by using a hash-based search. The current and left CTUs are used as a search area since the directions are 2-D for  $2N \times 2N$  and  $2N \times N$  PUs and 1-D for  $N \times 2N$  PUs. In the case of hash-based search, each  $2N \times 2N$  PU has a 16-bit hash key, which is estimated by the SC encoder. The current PU searches the matched block through the PUs that have the same key, and that by examining the hash table that contains the hash keys and the BVs for each hash key. The hash key is formulated from the DC values of the four  $4 \times 4$  blocks of each  $8 \times 8$  PU and the gradient of the whole PU. PLT mode is activated for the CUs which are less than  $64 \times 64$ . After completing the determination of optimum mode and RD cost, the encoder checks the CU size and goes to the next depth if the CU size is larger than  $8 \times 8$ . The encoder repeat this process again for higher depths. For  $8 \times 8$  CUs, the encoder stops the splitting process since  $8 \times 8$  size is the smallest size.

According to the analysis we have done in [22] by using the software test model HM16.7+SCM-6, hereafter SCM-6 for simplicity, the introduction of SC tools over conventional Intra mode increases the encoding time, since by disabling SC tools, the encoding time decreased by 58.27% on average while the performance is degraded clearly. Therefore, it is important to ignore the testing of unnecessary modes to save time and preserving the quality. Next section will discuss the proposed scheme.

### III. PROPOSED DECISION RULE BASED FAST INTRA CODING

In this research, we aim to accelerate the SC encoding process by utilizing a fast mode decision algorithm along with an early partitioning termination algorithm, and that to stop the execution of unnecessary mode testing. The skipping of SC tools is achieved by placing decision blocks before the run of the normal IBC and PLT modes to give decisions whether these modes would be bypassed or not. To decide if current CU could be partitioned or not, we suggest to insert a decision block after the mode testing which decides if the encoder goes to the higher depth or not for  $16 \times 16$  CUs or larger. The algorithms that represent the decision blocks for fast mode decision and fast CU decision are called Mode Skipping (MS) and Early Pruning Termination (EPT), respectively. The details about the features selection, the training methodology, and the mode decision of our proposed scheme will be discussed in next subsections.

#### A. FEATURES SELECTION

For efficient fast mode decision in MS algorithm, the candidate features should be suitable to differentiate between the different modes in Intra profile. In our scheme, the deactivation of SC tools is required, so, the Classification Parameter  $CP$ , as appeared in [29], was chosen as a feature for the training, and that because it has a high selectivity as reported in our previous work [30]. The Classification Parameter ( $CP$ ) is estimated as in [29]:

$$CP = \begin{cases} 0, & \text{if } CN = 1. \\ \frac{L_r}{CN-1} & \text{otherwise.} \end{cases} \quad (1)$$

where the Color Number  $CN$  represents the number of distinct colors inside the CU.  $L_r$  characterizes the range value between the maximum  $L_{max}$  and minimum  $L_{min}$  luma values and it is determined as

$$L_r = L_{max} - L_{min} \quad (2)$$

Figure 1 illustrates the percentages of NC and SC CUs versus  $CP$ . The data were extracted from the first 10 frames of the sequences that are recommended in [31]. The details of the recommended sequences are shown in Table 1 including the resolution, the number of encoded frames, and number of frames per second (fps). From figure 1, we can conclude that at the lower  $CP$  values, most of the CUs are NC type while at higher  $CP$  values, almost all CUs are encoded by using SC

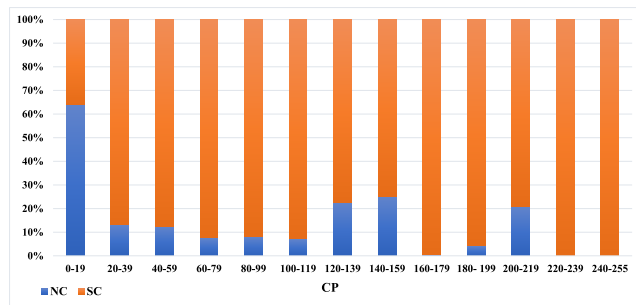


FIGURE 1. Percentages of NC and SC CUs versus CP for  $8 \times 8$  CUs.

tools, especially for  $CP$  more than or equal 220. PLT mode is more sensitive to the number of colors inside the CU, since the CU that has a high number of color requires more indices, which increases the coding cost [32]. Consequently,  $CN$  is considered as a training feature. Furthermore, the best RD costs before the run of the IBC and PLT modes are taken as features, and that because a small RD cost indicates that the previous modes are sufficient for the mode decision process.

In the case of the EPT, we should select proper features to discriminate between the CUs that should be split and the CUs that may be not partitioned without affecting the performance to save time. CU smoothness can characterize this behavior, since smooth blocks tend to be unpartitioned without affecting performance. CU variance  $var$  can be used to represent the CU structure complexity, which is estimated as:

$$var = \frac{1}{M \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} (L_{x,y} - L_{mean})^2 \quad (3)$$

where  $M$  and  $N$  are the height and the width of the CU.  $L_{mean}$  is the mean luminance value of the CU.  $L_{x,y}$  is the luminance value at location  $(x,y)$ . To study the relation between the CU variance and the normal splitting process, the first 10 frames of eight selected sequences were encoded, then the data were extracted for analysis. The chosen sequences are “sc\_SlideShow”, “sc\_programming”, “sc\_map”, “ChineseEditing”, “sc\_desktop”, “MissionControlClip3”, “Basketball\_Screen”, and “Kimono1”. The extracted data were partitioned into two groups. The first group is called “Not split”. In which, the RD cost of each CU is smaller than or equal to the sum of RD costs of its sub-CUs. While the second group, which is denoted as “Split”, is the opposite case. Figure 2 shows the histograms that represent the number of coded CUs versus the  $Var$  value for  $16 \times 16$ ,  $32 \times 32$ , and  $64 \times 64$  CU sizes, and that at each group, since “Not split” group is shown in figures 2a to 2c and “split” group is shown in figures 2d to 2f. From figure 2, we can show that the majority of the CUs under “Not split” span over a small range of  $Var$  value in contrary to “Split” category. Thus,  $Var$  is a good candidate feature for EPT algorithm. Figure 3 depicts similar histograms but for the best RD cost rather than variance value. From figure 3, we can show that the CUs belong to “Split” group extend more along the best RD cost range compared with the “Not

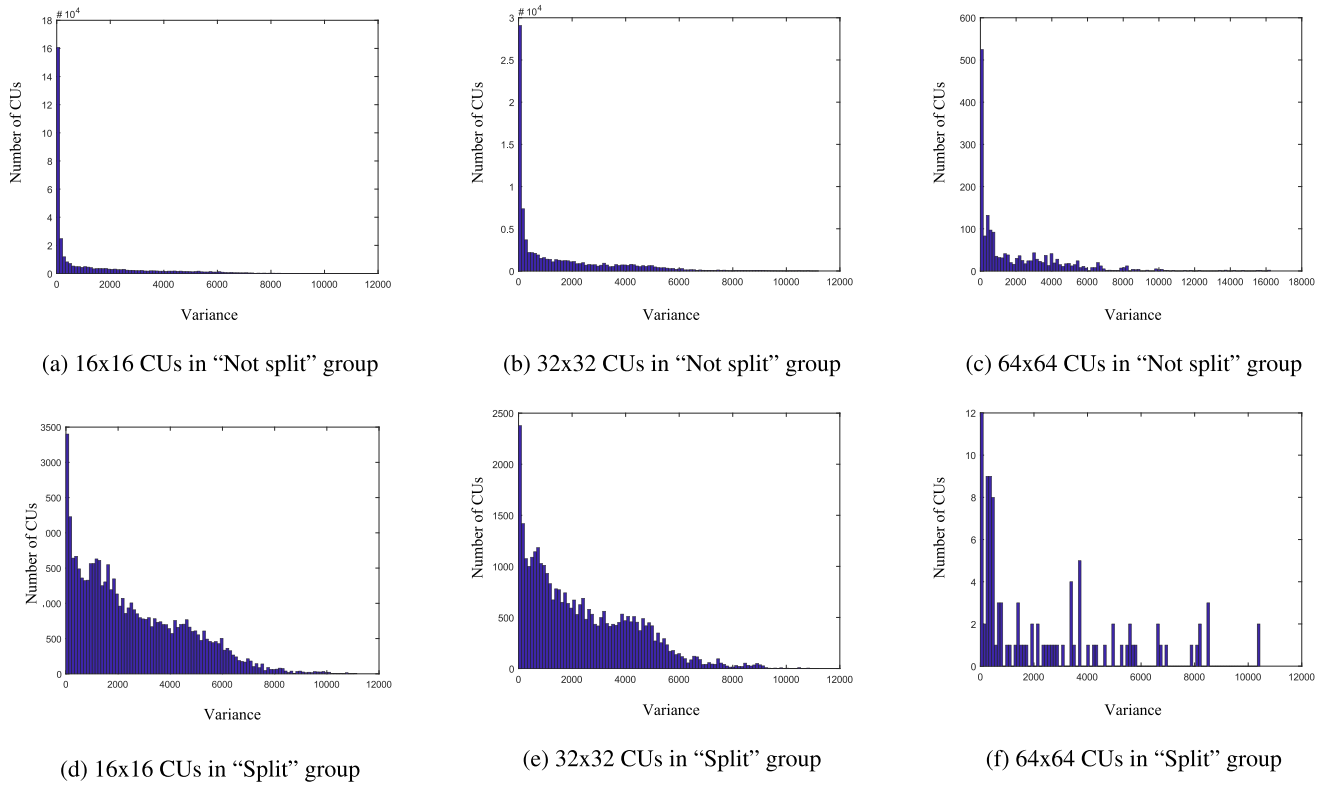


FIGURE 2. The number of CUs versus the variance.

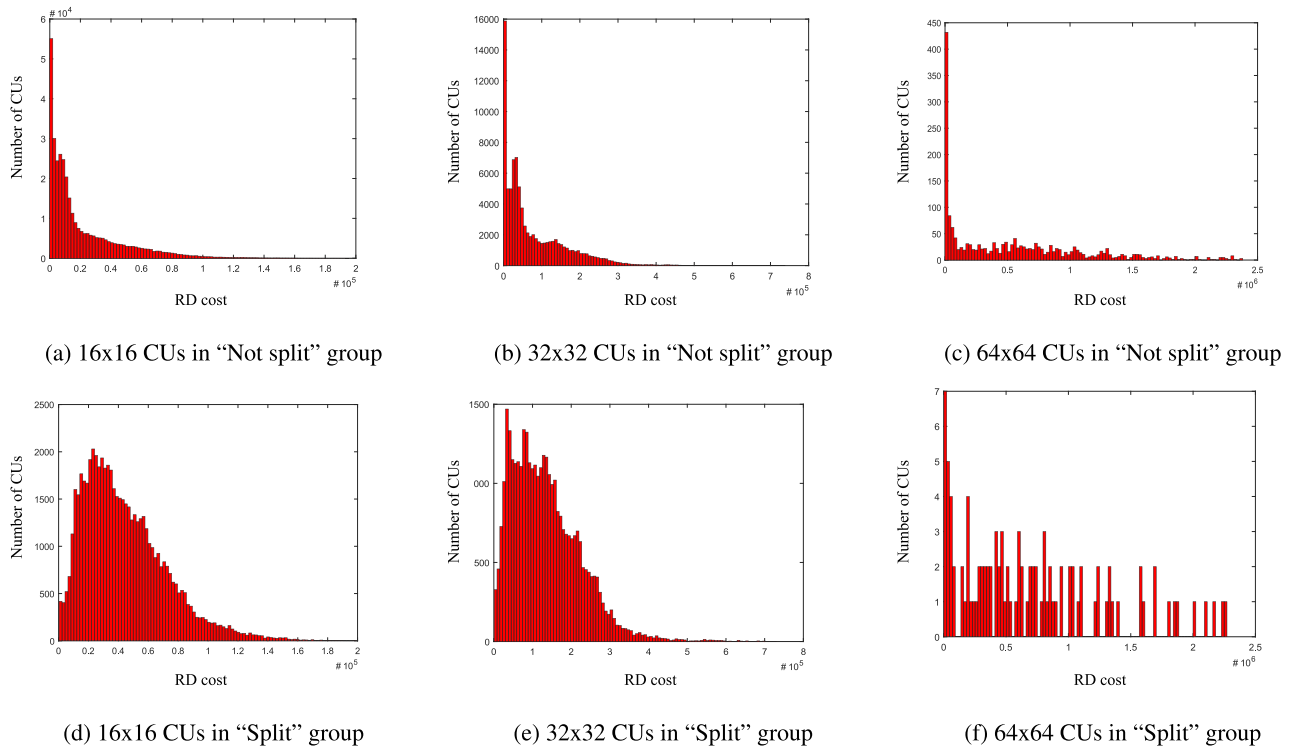


FIGURE 3. The number of CUs versus the best RD cost.

split” CUs. So, the best RD cost was chosen as a candidate feature to train the decision models. In conclusion, in the case of MS algorithm, *CP* and the best RD cost before IBC were selected for the decision block that is located before

IBC. *CN* and the best RD cost before PLT mode were chosen for the decision block that is inserted before the PLT mode. For EPT algorithm, *Var* and the best RD cost were chosen.

TABLE 1. Sequences that are recommended in [31].

Sequences	Resolution	Number of encoded frames	FPS
sc_web_browsing	1280x720	300	30
sc_map	1280x720	600	60
sc_programming	1280x720	600	60
sc_desktop	1920x1080	600	60
sc_SlideShow	1280x720	500	20
sc_flyingGraphics	1920x1080	300	60
sc_console	1920x1080	600	60
Basketball_Screen	2560x1440	300	60
MissionControlClip2	2560x1440	300	60
MissionControlClip3	1920x1080	600	60
sc_robot	1280x720	300	30
Kimono1	1920x1080	120	24
ChineseEditing	1920x1080	600	60

```

@RELATION SC_SKIP

@ATTRIBUTE CF NUMERIC
@ATTRIBUTE RDcost NUMERIC

@ATTRIBUTE class {0,1}

@DATA
219,63.198,0
0,57.452,0
82,476.66,0
82,396.71,0
219,68.943,0
0,51.707,0
219,57.452,0
0,45.962,0
82,91.198,0
0,51.707,0

```

FIGURE 4. A part of ARFF file.

## B. TRAINING METHODOLOGY

To train the predictive models of the decision blocks, the fast and popular machine learning classifier (JRip) [33] was used, which is a type of Decision Rules (DR) classifiers. It finds a set of rules that totally cover the members of classes. The generated rules are expressed as decision models. Comparing with other machine learning techniques such as Neural Networks (NN) and Supported Vector Machine (SVM) that require intensive mathematical computations, the decision models of DR are simple and can be converted to logic statements easily (IF-AND-OR). To extract data samples for the training, eight sequences from the recommended sequences were chosen. Table 2 tabulates information about the sequences that are used for training including the number of trained frames. The Waikato Environment for Knowledge Analysis (WEKA) [34], version 3.9.4, was used to build the decision rules. WEKA is an open-source software includes several machine learning algorithms. In the beginning of the training phase, the attributes and the classes are prepared into Attribute-Relation File Format (ARFF) file format, which is the input file to the WEKA tool. ARFF file contains the header, attributes deceleration, and raw data, which represent attributes and classes. Figure 4 shows a part of ARFF file. The training was done by using 10-fold cross-validation process. Finally, DR were generated to be incorporated into the SCM-6 reference software.

Table 3 shows the prediction accuracy of the training in each algorithm, and that at Quantization Parameter (QP) equals 22. In the case of MS algorithm, the models that are

used to skip SC tools, including IBC mode and the PLT mode, are called Screen Content Skipping (SCS). While the models that are located before the PLT mode are called PLT Skipping (PLTS). From Table 3, we can note that the prediction accuracy is high, especially for SCS models. SCS models are not trained for  $64 \times 64$ , since normal IBC and PLT are not carried out for this size. Similarly, in the case of PLTS for  $64 \times 64$  and  $32 \times 32$  sizes. Because  $8 \times 8$  CU size is the smallest size, the EPT algorithm is not conducted for  $8 \times 8$  CUs.

## C. MODE DECISION OF THE PROPOSED SCHEME

The mode decision flow of the SC encoder containing the proposed modifications is shown in figure 5. At the beginning, three flags are initialized to zero value. The flags are *skip\_SC*, *skip\_PLT*, and *skip\_split*. The first two flags are used in the MS decision blocks, while the last one is utilized in the EPT decision block. Then, *Var* attribute is calculated for the CUs more than  $8 \times 8$  pixels. *CP* attribute is estimated for the CUs less than  $64 \times 64$  to be used in the mode skipping as mentioned before. Before testing IBC mode, DR-based decision block gives a decision to skip SC modes or not. If it decides to skip IBC and PLT modes, *skip\_SC* is set to one. This model gives decisions for  $16 \times 16$  and  $8 \times 8$  CU sizes. Similarly, a decision is taken by a decision block resided before the execution of PLT mode to activate or bypass PLT mode. *skip\_PLT* is set to one if the block decides to skip PLT mode. It is used at CU sizes of  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$ . After finishing the modes execution and determining the best mode so far, decision block gives a decision to stop the pruning process or not, *skip\_split* is equal to one if the decision of early splitting termination is taken. If *skip\_split* equals one or if the CU size is  $8 \times 8$ , the encoder terminates the CU partitioning. Otherwise, it goes to the higher depth level to repeat the mode testing again.

## IV. RESULTS AND DISCUSSION

In order to evaluate the proposed scheme, we have implemented the scheme in the test model software SCM-6. The implemented version was simulated under the AI configuration profile. The QPs of 22, 27, 32, and 37 were used. In the evaluation steps, each proposed algorithm was evaluated in



TABLE 2. Sequences that are used for training.

Sequences	Number of frames used for training
sc_web_browsing	20
sc_programming	20
sc_flyingGraphics	10
sc_console	10
Basketball_Screen	10
sc_robot	20
Kimono1	10
ChineseEditing	10

TABLE 3. Prediction accuracy at QP22.

CU size	MS		EPT
	SCS	PLTS	
8x8	93.63%	90.15%	-
16x16	95.53%	82.10%	91.16%
32x32	93.46%	-	88.67%
64x64	-	-	86.35%

In addition to evaluating the integration of them all together, then, the proposed scheme has been compared with existing machine learning-based approaches. The evaluation metrics are the Bjontegaard Delta bit-rate (BD-R) [35] and the Time Savings TS, and that with respect to the anchor SCM-6. TS is described as

$$TS = \left\{ \frac{T_{anc} - T_{mod}}{T_{anc}} \right\} \times 100\% \quad (4)$$

where  $T_{anc}$  is the encoding time of the anchor version of SCM-6,  $T_{mod}$  is the encoding time of the SCM-6 containing the modifications. The positive value of TS means higher time saving.

Table 4 tabulates the evaluation results of the proposed algorithms EST and MS plus the EPT+MS. In addition, it includes comparisons with other machine learning-based approaches. From this table, EPT achieves 16.41% encoding time reduction with BD-R increment of 0.52% on average. The sequences that have the highest encoding time reduction in EPT algorithm are “sc\_SlideShow” and “MissionControl-Clip2” where TS values are recorded as 43.6% and 26.87%, respectively. While the BD-R increment values are 0.39% and 1.83%, respectively, and that because most of CUs in these sequences have a high degree of smoothness, which are likely to have small variance and RD cost. “sc\_flyingGraphics”, “sc\_console”, and “ChineseEditing” sequences show the smallest TS values for EPT algorithm with 8%, 9.85%, and 9.34%, respectively. These sequences contain complex shapes which require to be partitioned into smaller sizes to encode small details efficiently. Figure 6 depicts a scene from “sc\_flyingGraphics” sequence showing complex and sharp shapes.

In the case of MS, this algorithm reduces the encoding time by 29.27% on average while the BD-R increases to 0.89%. “sc\_robot” and “Kimono1” sequences show the highest encoding time reduction by 54.96% and 49.71%, respectively, with 1.01% and 0% BD-R increment, respectively. Most of the CUs in “sc\_robot” and “Kimono1” have small CP value, which can be efficiently encoded by Intra mode

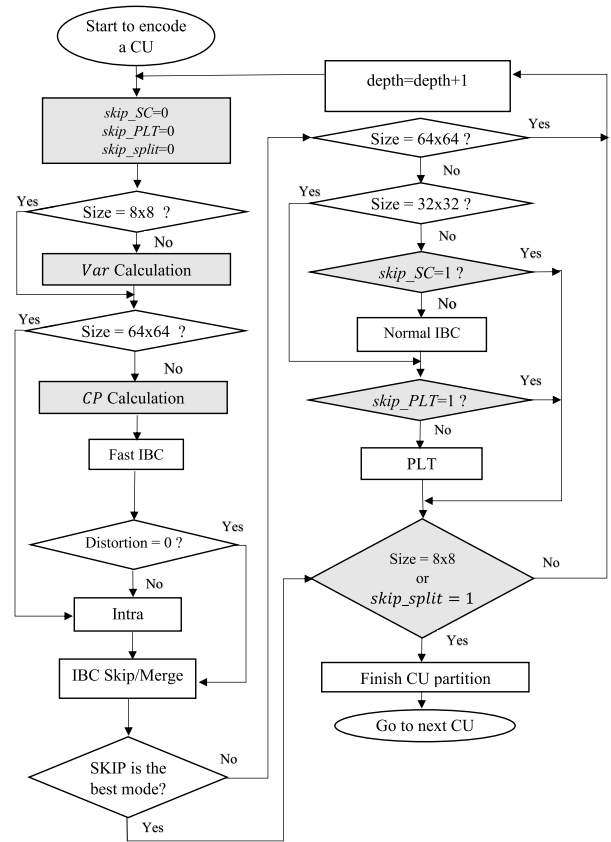


FIGURE 5. The flow chart of proposed scheme. The modifications are highlighted in gray color.

without needing to be checked by the SC modes. In addition, “sc\_SlideShow” records a high TS value, which is 43.85%, while the BD-R increases to 0.46%, and that because it is rich in flat regions, which means zero CP value. These flat regions could be handled by Intra mode well.

By combining MS and EPT algorithms, the scheme can decrease the encoding time by 37.84% on average and up to 63.63%. While the average BD-R increment is only 1.34%. Proposed scheme is compared with other existing ML-based approaches. Schemes in [24] and [23] were selected for comparisons, which are based on the RF and DT techniques, respectively. To achieve fair comparisons, approaches [24] and [23] were re-implemented in SCM-6 as our scheme, then they have been simulated under the same conditions. In [24] and [23], trained decision blocks have been inserted before the execution of the different coding modes. TS and BD-R results are shown in Table 4. From Table 4, it can be seen that our scheme has the lowest BD-R increment among existing schemes, since TS value is very close to that in [23]. Ref. [24] is the slowest approach between them, since the TS value is 29.88% with 1.37% BD-R increment. “sc\_robot” sequence achieves the best TS value in all approaches since most of its CUs can be characterized as NC type, which can be encoded by Intra mode with small RD cost as mentioned before, which can be handled by the trained blocks located before the SC modes.

TABLE 4. Evaluation results.

Sequences	[24]		[23]		EPT		MS		EPT+MS	
	BD-R (%)	TS (%)	BD-R (%)	TS (%)	BD-R (%)	TS (%)	BD-R (%)	TS (%)	BD-R (%)	TS (%)
sc_web_browsing	0.89	28.87	1.84	40.62	0.24	13.33	0.4	21.99	0.89	27.84
sc_map	0.84	25.65	1.72	29.55	0.37	16.02	1.38	28.03	1.73	38.67
sc_programming	1.81	24.98	1.76	37.92	0.5	12.57	0.63	20.5	1.03	29.81
sc_desktop	1.1	26.68	1.29	37.75	0.79	10.63	0.5	12.2	1.07	17.73
sc_SlideShow	4.42	25.84	4.44	37.47	0.39	43.6	0.46	43.85	1.01	55.2
sc_flyingGraphics	0.7	22.2	1.8	31.65	0.5	8	1.51	20.38	1.86	25.9
sc_console	0.86	27.21	1.16	39.97	1.23	9.85	0.82	19.93	1.64	24.12
Basketball_Screen	1.74	26.62	1.83	39.5	0.19	15.63	0.95	28	1.03	36.35
MissionControlClip2	1.88	28.34	2.71	42.4	1.83	26.87	2.42	33.88	3.94	50.63
MissionControlClip3	1.47	30.03	1.86	39.82	0.35	17.98	0.89	27.83	1.17	39.52
sc_robot	0.87	49.91	1.79	53.7	0.17	14.47	1.01	54.96	1.2	63.63
Kimono1	0.05	42.4	0.72	45.5	0.11	15.1	0	49.71	0.14	60.39
ChineseEditing	1.2	29.74	0.68	32.5	0.1	9.34	0.62	19.34	0.7	22.18
average	1.37	29.88	1.81	39.1	0.52	16.41	0.89	29.27	1.34	37.84
Number of estimated features	9		9		3					

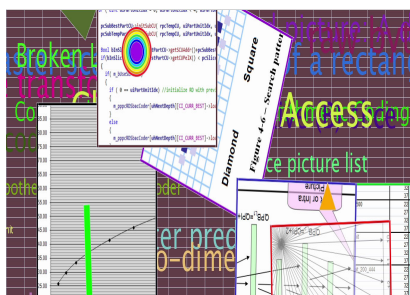


FIGURE 6. A picture from “sc\_flyingGraphics” sequence containing complex regions.

When it comes to the number of estimated features, EPT+MS has to estimate 3 features, which is the smallest. While 9 features have to be estimated in [24] and [23], and that after ignoring the estimation of the RD costs and flags because these features are found in SCM-6 by default. Furthermore, [24] has to access neighboring CUs to get coding information. This advantage makes our scheme more applicable to be used in the real video coding systems.

V. CONCLUSION

In this paper, the DR machine learning technique has been used to boost up the speed of SC encoders. Two algorithms have been proposed. MS, which is the first algorithm, is used to skip running of IBC or IBC and PLT modes, and that with accordance to decisions that are taken by trained models located before the running of IBC and PLT modes. The second algorithm, which is called EPT, is used to make fast CU size determination. The encoder has to estimate 3 features only to train the models. The simulation results show that the presented scheme gives 37.84% encoding time reduction while the BD-Rate increases by 1.34%, which outperforms other machine learning-based approaches.

ACKNOWLEDGMENT

The authors would like to thank E-JUST Center at Kyushu University and Egypt-Japan University of Science and

Technology for the continuous support. They would also like to thank the authors from [23] and [24] for the provision of source codes.

REFERENCES

- [1] J. Xu, R. Joshi, and R. A. Cohen, “Overview of the emerging HEVC screen content coding extension,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50–62, Jan. 2016.
- [2] S. Liu, X. Xu, S. Lei, and K. Jou, “Overview of HEVC extensions on screen content coding,” *APSIPA Trans. Signal Inf. Process.*, vol. 4, pp. 1–12, Aug. 2015.
- [3] W.-H. Peng, F. G. Walls, R. A. Cohen, J. Ostermann, A. MacInnis, T. Lin, and J. Xu, “Overview of screen content video coding: Technologies, standards, and beyond,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 393–408, Dec. 2016.
- [4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [5] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, “Intra coding of the HEVC standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.
- [6] M. Budagavi and D.-K. Kwon, “Intra motion compensation and entropy coding improvements for HEVC screen content coding,” in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 365–368.
- [7] W. Zhu, W. Ding, J. Xu, Y. Shi, and B. Yin, “2-D dictionary based video coding for screen contents,” in *Proc. Data Compress. Conf.*, Mar. 2014, pp. 43–52.
- [8] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.-K. Wang, and M. Karczewicz, “Intra block copy in HEVC screen content coding extensions,” *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp. 409–419, Dec. 2016.
- [9] L. Guo, W. Pu, F. Zou, J. Sole, M. Karczewicz, and R. Joshi, “Color palette for screen content coding,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5556–5560.
- [10] X. Xiu, Y. He, R. Joshi, M. Karczewicz, P. Onno, C. Gisquet, and G. Laroche, “Palette-based coding in the screen content coding extension of the HEVC standard,” in *Proc. Data Compress. Conf.*, Apr. 2015, pp. 253–262.
- [11] W. Pu, M. Karczewicz, R. Joshi, V. Seregin, F. Zou, J. Sole, Y.-C. Sun, T.-D. Chuang, P. Lai, S. Liu, S.-T. Hsiang, J. Ye, and Y.-W. Huang, “Palette mode coding in HEVC screen content coding extension,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 420–432, Dec. 2016.
- [12] Y. Zhang, C. Zhang, and R. Fan, “Fast motion estimation in HEVC inter coding: An overview of recent advances,” in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Honolulu, HI, USA, Nov. 2018, pp. 49–56.

- [13] D.-K. Kwon and M. Budagavi, "Fast intra block copy (IntraBC) search for HEVC screen content coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Melbourne, VIC, Australia, Jun. 2014, pp. 9–12.
- [14] Q. Hu, X. Zhang, Z. Shi, and Z. Gao, "Neyman-pearson-based early mode decision for HEVC encoding," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 379–391, Mar. 2016.
- [15] W. Zhu, W. Ding, J. Xu, Y. Shi, and B. Yin, "Hash-based block matching for screen content coding," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 935–944, Jul. 2015.
- [16] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Hash based fast local search for intra block copy (IntraBC) mode in HEVC screen content coding," in *Proc. APSIPA ASC*, Hong Kong, Dec. 2015, pp. 396–400.
- [17] S.-H. Tsang, W. Kuang, Y.-L. Chan, and W.-C. Siu, "Fast HEVC screen content coding by skipping unnecessary checking of intra block copy mode based on CU activity and gradient," in *Proc. APSIPA ASC*, Jeju, South Korea, Dec. 2016, pp. 1–5.
- [18] H. Zhang, Q. Zhou, N. Shi, F. Yang, X. Feng, and Z. Ma, "Fast intra mode decision and block matching for HEVC screen content compression," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Shanghai, China, Mar. 2016, pp. 1377–1381.
- [19] M. Zhang, Y. Guo, and H. Bai, "Fast intra partition algorithm for HEVC screen content coding," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Valletta, Malta, Dec. 2014, pp. 390–393.
- [20] J. Lei, D. Li, Z. Pan, Z. Sun, S. Kwong, and C. Hou, "Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding," *IEEE Trans. Broadcast.*, vol. 63, no. 1, pp. 48–58, Mar. 2017.
- [21] S.-H. Tsang, W. Kuang, Y.-L. Chan, and W.-C. Siu, "Reduced-complexity intra block copy (IntraBC) mode with early CU splitting and pruning for HEVC screen content coding," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 269–283, Feb. 2019.
- [22] E. Badry, K. Inoue, and M. S. Sayed, "Decision tree models and early splitting termination in screen content extension of high efficiency video coding," *IEEE Access*, vol. 8, pp. 143437–143452, 2020.
- [23] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Machine learning-based fast intra mode decision for HEVC screen content coding via decision trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1481–1496, May 2020.
- [24] S.-H. Tsang, Y.-L. Chan, and W. Kuang, "Mode skipping for HEVC screen content coding via random forest," *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2433–2446, Oct. 2019.
- [25] W. Kuang, S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast mode decision algorithm for HEVC screen content intra coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2473–2477.
- [26] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Online-learning-based Bayesian decision rule for fast intra mode and CU partitioning algorithm in HEVC screen content coding," *IEEE Trans. Image Process.*, vol. 29, pp. 170–185, 2020.
- [27] C. Huang, Z. Peng, F. Chen, Q. Jiang, G. Jiang, and Q. Hu, "Efficient CU and PU decision based on neural network and gray level co-occurrence matrix for intra prediction of screen content coding," *IEEE Access*, vol. 6, pp. 46643–46655, 2018.
- [28] W. Kuang, Y.-L. Chan, and S.-H. Tsang, "Low-complexity intra prediction for screen content coding by convolutional neural network," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, Oct. 2020, pp. 1–5.
- [29] M. Zhang, S. Wang, and B. Li, "Selective motion estimation strategy based on content classification for HEVC screen content coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2463–2467.
- [30] E. Badry, A. Shalaby, and M. S. Sayed, "Intra mode decision acceleration for HEVC screen content coding," in *Proc. 7th Int. Japan-Africa Conf. Electron., Commun., Comput. (JAC-ECC)*, Dec. 2019, pp. 92–95.
- [31] H.-P. Yu, R. Cohen, K. Rapaka, and J.-Z. Xu, *Common Test Conditions for Screen Content Coding*, document JCTVC-X1015, Geneva, Switzerland, May 2016.
- [32] E. Badry, A. Shalaby, and M. S. Sayed, "Fast algorithm with palette mode skipping and splitting early termination for HEVC screen content coding," in *Proc. IEEE 62nd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2019, pp. 606–609.
- [33] A. Rajput, R. P. Aharwal, M. Dubey, S. Saxena, and M. Raghuvanshi, "J48 and JRIP rules for e-governance data," *Int. J. Comput. Sci. Secur.*, vol. 5, no. 2, p. 201, 2011.
- [34] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
- [35] G. Bjontegaard, *Calculation of Average PSNR Differences Between R-D Curves*, document VCEGM33, ITU-T VCEG, Austin, TX, USA, 2001.



**EMAD BADRY** (Member, IEEE) received the B.Sc. degree in electronics and communication engineering from Suez Canal University, Ismailia, Egypt, in 2010, and the M.Sc. and Ph.D. degrees from the Egypt-Japan University of Science and Technology (EJUST), New Borg El Arab, Egypt, in 2018 and 2021, respectively. He was a Special Research Student with Kyushu University, Japan, from 2019 to 2020. He is currently an Assistant Professor with the Department of Electrical Engineering, Faculty of Engineering, Suez Canal University. His current research interests include video coding, machine learning, and digital circuits design.



**KOJI INOUE** (Member, IEEE) received the B.E. and M.E. degrees in computer science from the Kyushu Institute of Technology, Japan, in 1994 and 1996, respectively, and the Ph.D. degree from the Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, in 2001. In 1999, he joined Halo LSI Design & Technology, Inc., NY, USA, as a Circuit Designer. He is currently a Professor with the Department of Advanced Information Technology, Kyushu University. His research interests include power-aware computing, high-performance computing, secure computer systems, 3D microprocessor architectures, multi/many-core architectures, nanophotonic computing, and superconductor computing.



**MOHAMMED SHARAF SAYED** (Member, IEEE) received the B.Sc. degree in electronics and communications engineering from Zagazig University, Zagazig, Egypt, in 1997, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Calgary, Calgary, Canada, in 2003 and 2008, respectively. He is currently a Professor with the Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology, Egypt. His research interests include digital systems design, embedded systems, system-on-chip, machine vision, video coding, and wireless body area networks. He received several awards both in Canada and in Egypt. He has one U.S. patent and seven contributions to the ITU/MPEG video coding standards, two of them included in the H.264 standard's reference hardware model. He authored or coauthored more than 90 international publications. He has managed and participated in several national/international research and development projects. He acted as a reviewer for several IEEE transactions and other journals.

...