Defence Science Journal, Vol. 74, No. 2, March 2024, pp. 270-277, DOI : 10.14429/dsj.74.18780

High-Speed Coding Unit Depth Identification Based on Texture Image Information Using SVM

Pattimi Hari* and Kota Naga Srinivasarao Batta

National Institute of Technology Warangal, Telangana-506004, India.

**E-mail: harinitw@student.nitw.ac.in*

ABSTRACT

The High-Efficiency Video Coding (HEVC/H.265) standard was created by a group called the Joint Collaborative Team on Video Coding (JCT-VC) and was released in 2013.In HEVC Rate-Distortion Optimization (RDO) is employed for partitioning the coding tree units into coding units recursively due to this the computational complexity increases. However, changing from macroblocks to coding tree units the compression efficiency increases and increases the encoding of the video sequence. This paper presents a Support Vector Machine (SVM)-based method for finding the fastest coding unit division in intra-prediction HEVC without compromising compression efficiency. All partitions of CTU are assessed using five characteristics: Root Mean Square Error (RMSE), Sub CU Complexity Difference (SCCD), Standard Deviation (SD), Directional Complexity (DC), and Quantization Parameter (QP) to optimize the intra-prediction of HEVC in all intra-configurations. The simulation results reveal combining directional complexity and standard deviation yields a more accurate classification. SVM was used to separate split-unsplit data, and the conventional rate-distortion optimisation technique was applied to separate samples that were difficult to separate. The results show that the encoding process has been completed 67.44% faster, albeit with a marginal increase in bit rate.

Keywords: CTU; DC; HEVC; RMSE; SCCD; SD; SVM

NOMENCLATURE

PU	: Prediction Unit
TU	: Transform Unit
CB	: Coding Block
CTU	: Coding Tree Unit
CU	: Coding Unit
RDO Rate	: Distortion Optimization
Y	: Luminance
C_b	: Blue Chroma
C_r	: Red Chroma
HD	: High Definition
UHD	: Ultra High Definition
S_k	: Sobel Operator

1. INTRODUCTION

High-Efficiency Video Coding (HEVC) is the most recent video coding standard. It was introduced in 2013 and was developed by the Joint Collaborative Team on Video Coding (JCT-VC). HEVC is an improvement over its predecessor, H.264/AVC, due to its ability to decrease data requirements by 50% while maintaining video quality. By utilising a variety of techniques to enhance video quality while preserving data, HEVC increases the computational complexity of the encoder. The following are some of the novel functionalities introduced by HEVC to improve coding efficiency:

- A flexible quadtree structure.
- Increased intra-prediction.
- Adaptive sample offset.
- Advanced motion vector prediction.

Using these additional features in a real-time environment significantly increases computational complexity. The quadtree structure is used as a reference when dividing a video frame from a particular sequence into non-overlapping Coding Tree Units (CTU). The largest and smallest CUs in the CTU partition are 64×64 and 8×8, respectively[3]. Each 64×64 coding unit is subdivided into four sub-CUs recursively until the 8×8 coding unit hasreached. Fig. 1 depicts an intra-prediction coding tree unit partition for HEVC. The CU, often called the CTU, is 64×64 and has a depth of 0. Each sub-CU is 32×32 and has a depth of one. The CU depth is continuously increasing and being divided into smaller CUs to improve the performance of an image. For instance, the size is 16×16, and the CU depth of the letters a, b, g, j, o, and p is 2. CUs labeled as c, d, e, f, k, l, m, and n are 8x8 in size and maximum depth is 3.

CU is composed of the three Coding Blocks (CBs) Luminance (Y), Blue Chroma (Cb), and Red Chroma for a color image (Cr). CB is sufficient for determining prediction type, but it may need to be more significant to store Motion Vectors (MV) inter prediction or intra-prediction mode.The Prediction Unit (PU) for each CU must be determined[4-7]. Each PU has one of 35 distinct prediction modes. The best PU must be chosen and assigned to the corresponding CU. HEVC uses the Rate-

Received : 22 December 2022, Revised : 01 December 2023
Accepted : 04 December 2023, Online published : 26 March 2024

Distortion Optimization (RDO) method[8-12]. Each CU must be iteratively checked to identify which of the 35 prediction modes has the lowest rate-distortion cost. For a standard 64×64 block, one must look at RD cost to get prediction units for the whole block.

The present paper is organised in the following manner: Section 2 of the paper presents a comprehensive compilation of pertinent scholarly works. The discussion on the complexity of an image's texture is included in Section 3, while Section 4 provides a description of the machine-learning algorithms proposed for reducing the complexity of coding unit partitioning in all intra-HEVC configurations. The empirical results outlined in Section 5. The conclusion of our paper is presented in Section 6.

2. LITERATURE REVIEW

Over the last decade, numerous researchers have developed various algorithms and methods for faster determining the coding unit partitioning of an image. Some ways to divide up coding units are based on recursion, while others use online learning. Finding the coding unit partition with recursive-based methods is much harder and takes a lot more time than with online-based methods. In this section, some of the most recent online learning methods have beendiscussed.

Lee & Jeong[13] suggest a machine learning algorithm to quickly decide the depth of coding units in HEVC intra-coding. They use RDO to measure complexity. They employ K- nearest neighbor and Fischer linear discriminant analysis for efficient CU partitioning decision-making.

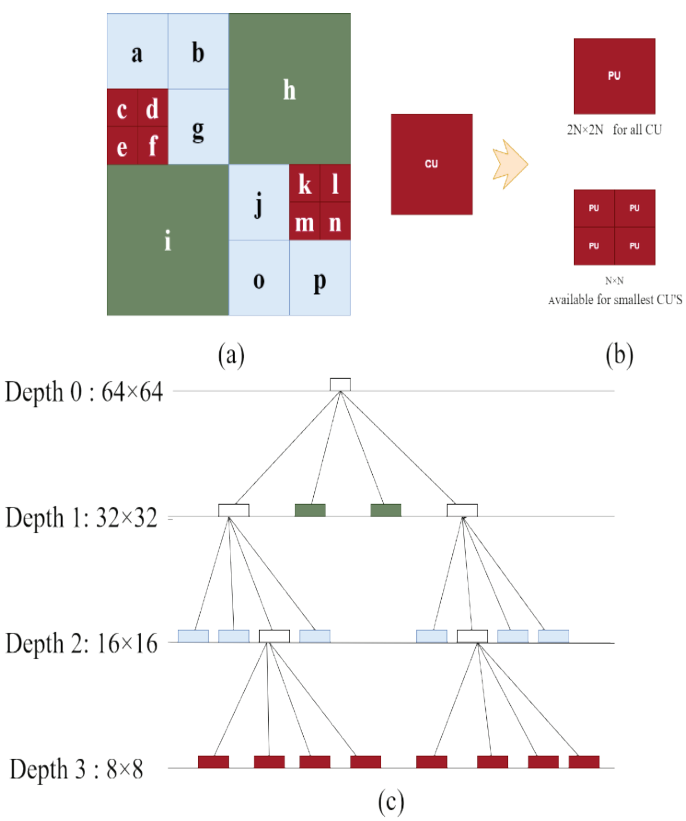

Figure 1. Partition of a coding tree unit. (a) The final partition of a coding tree unit. (b) CU and PU partition. (c) Quad-tree partition of a coding tree unit.

Bai & Yuan[14] propose a quickly decide the CTU decision technique for accelerating HEVC intra-prediction by using the Sobel operator and maximum absolute difference to remove texture complexity modes. However, the combination of these methods may result in the skipping of certain coding units during the process.

Li[15], *et al.* & Imen[16], *et al.* propose using CNN-based algorithms as an alternative to the recursive search used in RDO for coding unit partitioning. This strategy maintains compressionefficiency while significantlyreducing the time required to determine if all depths need to be split.

Guanwen Zhang[17], *et al.* suggest a deep learning-based approach for CU partitioning in HEVC intra-prediction. Their method utilizes a CNN to forecast whether a given CU (64×64, 32×32, or 16×16) should be divided, reducing the need for recursive search, and improving efficiency.

Kim & Ro[18] introduce a neural network architecture for predicting CU division in HEVC intra-coding. They develop a neural network database that encodes image and vector data to enhance prediction accuracy, enabling effective determination of CU partitioning depths.

Wang & Li[19] introduce a one-step method for dividing coding units and prediction units in HEVC intra-coding.They use a one-step decision network structure in their approach to predict CU partitioning. This helps reduce complexity and eliminates the need for repeated searching.

Bouaafia[20], *et al.* suggest a fast CU splitting strategy for HEVC intra-coding using a deep learning model. Their approach combines CNN-LSTM to predict CU splitting and decrease encoding complexity. The deep framework is trained and evaluated using an efficient dataset.

Bouaafia[21], *et al.* demonstrate a machine learning-based approach for coding unit division in inter-mode HEVC. They use an online SVM to simplify things and build a deep learning framework. This framework relies on a large training database to predict coding unit partitions.

Existing literature offers numerous models and methods for reducing complexity and bit rate in Coding Unit (CU) partitions. However, many of these techniques suffer from time-consuming computations, computational complexity, or limited bit rate reduction. Many people want faster and simpler methods for coding unit partitioning. This study suggests using a machine learning approach to fulfill this demand with a method that is not too complex.

The literature[13-21],suggests that various texture features, such as standard deviation, root mean square error, sub-CU complexity difference, directional complexity, and quantization parameter, should be employed to determine the homogeneity or diversity of CUs within CTUs during the CU splitting process. By incorporating these multiple features, a more comprehensive evaluation of CU complexity can be achieved, leading to improved CU partitioning decisions and potentially better compression efficiency.

3. MULTIPLE FEATURES FOR FINDING IMAGE TEXTURE COMPLEXITY

The texture features used for computing image complexity are classified into three categories: Class A (non-homogeneous), Class B (homogeneous), and Class C (recursive-based technique). This classification improves the accuracy of texture classification and reduces the computational load for coding unit depth calculations.

The image complexity is divided into two categories: direction information and texture information. The texture information is determined by three distinct characteristics:

3.1 Root Mean Square Error (RMSE)

It calculates the average gap between the pixel values of an image block and the values predicted for it. Higher RMSE indicates greater complexity and depth within the block. The RMSE of an image is calculated using equation Eqn. (1).

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(h(i,j)-\bar{h}(i,j))^2}{N}} \quad (1)$$

In this context, h (i, j) is the luminescent value of a pixel at location i, j. $\bar{h}\,(i,j)$ stands for the average luminescent of the entire block, and N represents the block's size.

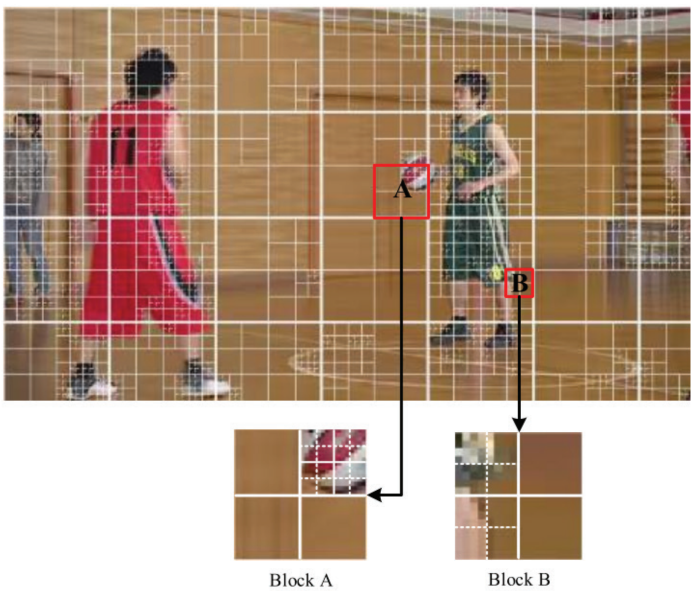

Figure 2. Example of coding unit partitions obtained for an image in basketball pass sequence.

3.2 Sub-CU Complexity Difference

SCCD measures the complexity difference between the sub-coding units within a block as shown in Fig. 2. The second sub-CU of block A is separated into smaller blocks because of the second sub-CU's complex texture. Block B is already broken into other blocks to show its complexity. The SCCD is higher when the block is more likely to split and lower when the sub-CU complexity is different. A higher complexity difference suggests a higher probability of splitting for the block. The SCCD of a block is calculated by Eqn. (2)

$$SCCD = \frac{1}{4}\sum_{i=0}^{3}(var_i - \overline{var})^2 \quad (2)$$

where, var_i is the ith sub-CU of the block's variance, and $\overline{var}$ is the total variance over all four sub-CUs.

3.3 Standard Deviation

SD calculates the dispersion of pixel values within an image block. Higher standard deviation indicates greater variation and complexity.The standard deviation is calculated using Eqn. (3).

$$SD = \sqrt{\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(h(i,j)-\bar{h}(i,j))^2}{N^2-1}} \quad (3)$$

where, N represents the size of the block, h (i, j) is the luminescent value of a pixel at coordinatesi, j, and $\bar{h}\,(i,j)$ is the average luminescence of the entire block.

Additionally, the Directional Complexity (DC) is computed using the Sobel operator to capture direction information. The angular Sobel operator estimates the gradient of the image and helps determine its complexity in different directions as shown in Table.1. Eqn. 4 and 5 helps to capture the directional information.

Table 1. Angular Sobel operators S_k at each pixel location (horizontal, vertical, 45^0, and 135^0).

1	2	1
0	0	0
-1	-2	-1

S_{hor}

1	0	-1
2	0	-2
1	0	-1

S_{ver}

0	1	2
-1	0	1
-2	-1	0

S_{45^0}

2	1	0
1	0	-1
0	-1	-2

S_{135^0}

$$G_k(i,j) = S_k \times A, k = (hor, ver, 45^0, 135^0)$$

$$A=\begin{bmatrix} g(i-1,j-1) & g(i-1,j) & g(i-1,j+1) \\ g(i,j-1) & g(i,j) & g(i,j+1) \\ g(i+1,j+1) & g(i+1,j) & g(i+1,j+1) \end{bmatrix} \quad (4)$$

$$Dcom = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(\begin{matrix} |g_{hor}(i,j)| + \\ |g_{ver}(i,j)| + |g_{45^0}(i,j)| + \\ |g_{135^0}(i,j)|) \end{matrix}\right) \quad (5)$$

where, g (i, j) is the luminescent value at that location, N is the block size, and Gk is the Sobel matrix.

The Quantization Parameter (QP) step is also considered, as it directly affects the bitrate and coding unit partitions in HEVC encoding. It is normalized using the Qstep value, which is calculated based on the QP.The quantization parameter is determined using Eqn. (6) as follows:

$$Q_{step} = 2^{\frac{QP-4}{6}} \quad (6)$$

If the QP increases by 1, Q_{step} varies by 12.5 per cent. Thus, it is observed that for a slight change in QP, Qstep changes drastically.

By considering these features, including RMSE, SCCD, SD, DC, and QP step, the proposed approach aims to effectively evaluate the texture and directional complexity of coding units for more accurate and efficient coding unit partitioning.

4. PROPOSED MACHINE LEARNING-BASED ALGORITHM FOR FAST CODING UNIT DEPTH DECISION IN HEVC INTRA-PREDICTION

The motivation behind this research is to explore SVM-based approaches for simplifying the HEVC intra-coding process. By employing online learning techniques, it is possible to reduce the complexity and computational load involved in determining coding unit depths, thus achieving faster encoding while maintaining satisfactory coding efficiency.

4.1 Support Vector Machine (SVM) Learning Algorithm for Finding Coding Unit Depth Values

Compared to deep learning techniques[22], the Support Vector Machine (SVM) algorithm requires less training time. SVMs can find the decision boundary using only the support vectors, which results in a smaller subset of the dataset being used for training. This is advantageous when accurate classification is achieved, as less data is needed for training.

SVM utilizes the kernel trick, a technique that projects data into higher dimensions, enabling the establishment of a decision boundary in n-dimensional space.This allows for the classification of data into different classes using an optimal hyperplane.

When predictability of the learning model's results is a concern, it is advisable to initially explore conventional machine learning methods before opting for deep learning models.

The proposed methodology as shown in Fig. 3 for identifying optimal coding unit (CU) depths in intra-prediction involves calculating the root mean square error, standard deviation, sub-coding units complexity difference and directional complexity for each feature of the coding tree units. To improve distribution properties and eliminate redundant information, it is recommended to normalize the values of the texture features. The normalization is achieved by applying a logarithm transformation Eqn. (7).

$$X_{new=\log_{10} X} \tag{7}$$

SVM is utilized to classify the normalized texture data. Simulation results have shown that SVM achieves higher accuracy in classifying image blocks compared to other classification algorithms like K-Nearest Neighbor (KNN), Logistic Regression (LR), Naive Bayes, etc.

In the proposed approach, the samples are divided into three classes (A, B, and C) using SVM[23]. Figure 4 demonstrates a two-class classification example, indicating that certain samples exhibit significant ambiguity. To reduce ambiguity, the proposed method employs a three-class classification, as depicted in Fig. 5.

Figure 5 illustrates the three-class classification by dividing the plot into classes A, B, and C. Class A samples are mostly split, while class B samples are predominantly un-split. On the other hand, class C samples consist of both split and un-split samples, making their classification more challenging. The difficulty lies in identifying the sample class for class C. This means that the original RDO cost method must be employed to determine whether samples from class C should be split.

In the proposed method, RDO cost is used to identify only class C samples, while SVM is employed to classify samples belonging to class A and class B. Due to the smaller number of samples in class C compared to classes A and B, this

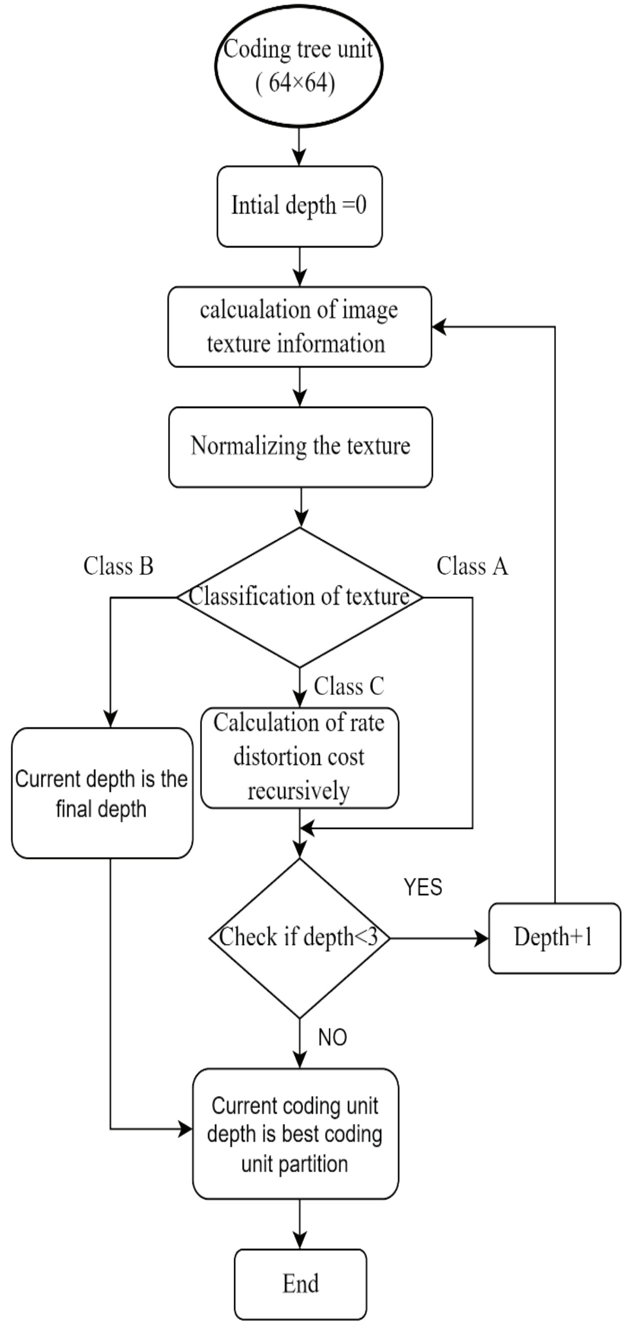

Figure 3. Proposed algorithm to calculate the CU depths.

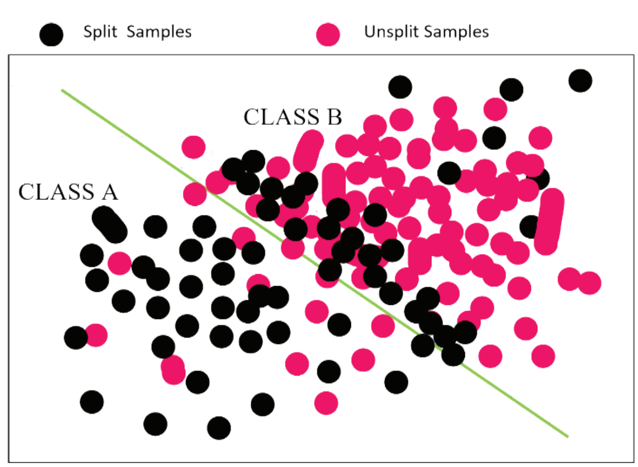

Figure 4. An example for 2 class classification afternormalizing.

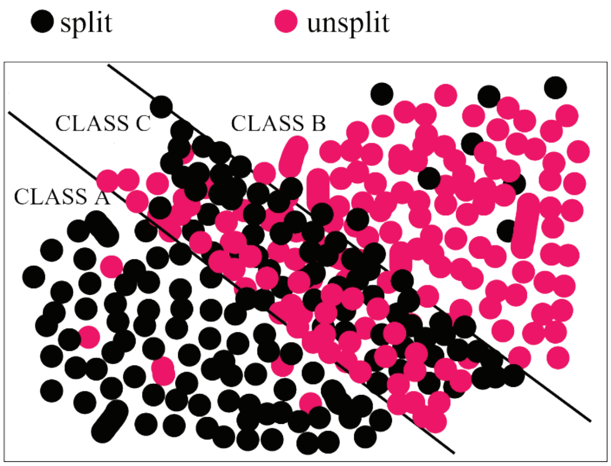

Figure 5. An example of three-class classification after normalizing.

approach decreases complexity and reduces the time required for classification.

The literature discusses the limitations of using a single feature for rate distortion optimization (RDO), which divides samples into different block sizes such as CTU (64×64), CU (32×32), and sub CU (16×16). By relying on a single feature, the information in the image is not fully utilized, leading to the loss of some information during reconstruction. Moreover, the accuracy and computation time for sample separation based on rate distortion using a single feature are poor, as indicated in Table 2.

Table 2. Accuracy comparison with a single feature.

Single-texture feature considered	64 × 64 Split/unsplit accuracy classification (%)	32 × 32 Split/unsplit accuracy classification (%)	16 × 16 Split/unsplit accuracy classification (%)
Rate-Distortion (RD) cost	87.5	77.4	69.0

Table 3. Accuracy comparison with two features

Multiple features considered	Accuracy with block sizes		
	64 × 64 (%)	32 × 32 (%)	16 × 16 (%)
Dcom and SCCD	92.78	87.11	82.86
Dcom and RMSE	91.12	87.15	84.30
Dcom and SD	**93.03**	**87.28**	**85.16**
SCCD and RMSE	92.60	87.08	83.51
SCCD and SD	92.30	86.35	83.49
RMSE and SD	91.81	87.20	83.32

To address these limitations, this paper proposes using multiple features to retrieve texture information from images. These features include RMSE, SD, SCCD, DC, and the quantization parameter, as shown in Table 3. Simulations are conducted with different combinations of these features as inputs to the SVM, and the accuracy is evaluated. It is observed that by incorporating multiple features, the accuracy is improved compared to using a single feature, albeit with a modest increase in computational complexity.

Table 4. Accuracy comparison with more than two features.

More than two features are considered	Accuracy with block sizes		
	64 × 64 (%)	32 × 32 (%)	16×16 (%)
Dcom, SCCD, and RMSE	91.29	85.33	84.75
Dcom, SCCD, and SD	90.08	84.14	84.68
RMSE, SCCD, and SD	89.76	86.43	82.44
RMSE, Dcom and SD	91.43	86.32	84.13
Dcom, SCCD, RMSE, and SD	92.33	86.43	85.04

Tables 3 and Table 4 present the results of simulations with combinations of 2, 3, and 4 features to reduce computational complexity. It is found that the accuracy achieved with combinations of two or more features is nearly identical. Therefore, two features are selected as inputs to the SVM for classification accuracy assessment. The directional complexity and standard deviation are chosen as the two input features, and they exhibit good accuracy in identifying the class to which each CTU belongs and estimating the depths of the bestCUs.

Figures 6, 7, and 8 illustrate the classification texture with the highest accuracy based on directional complexity and standard deviation for 64×64, 32×32, and 16×16 samples, respectively. The split samples are depicted in purple on the left side of these graphs, while the un-split samples are shown in yellow on the right side.

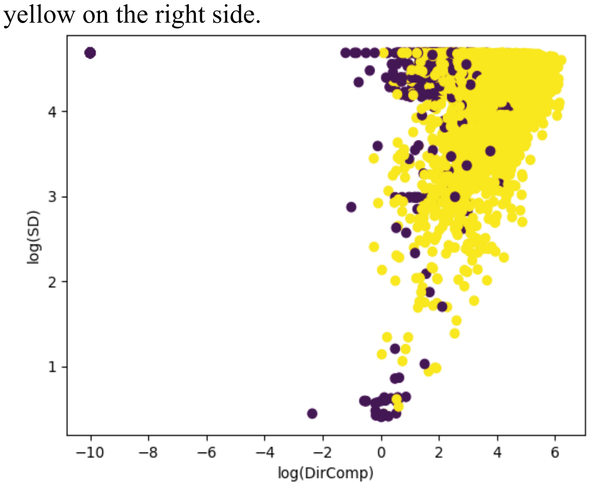

Figure 6. Dcom and SD for 64×64 samples.

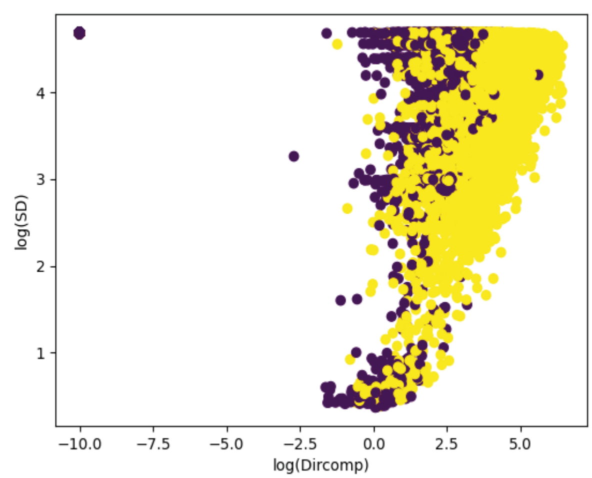

Figure 7. Dcom and SD for 32×32 samples.

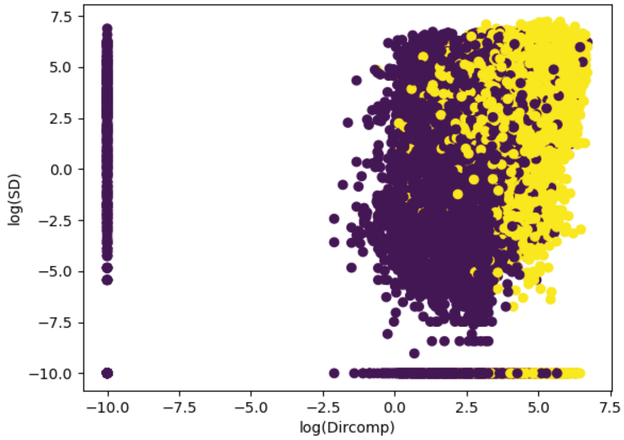

Figure 8. Dcom and SD for 16×16 samples.

Table 5. The suggested algorithm's performance is compared to HM15.0 in terms of performance using the support vector machine

Test Classes	Test Sequence	BR (%)	BP (%)	ΔT (%)
Class A 2560×1600	Traffic	1.7	0.3	65.41
	People On Street	1.3	0.3	67.20
Class B 1920×1080	Kimono	1.4	0.6	63.60
	Park Scene	2.3	0.8	61.10
Class C 1280×720	Kristen and Sara	1.9	0.5	65.48
	Four People	1.7	0.4	67.71
	Johnny	1.5	0.5	67.90
Class D 832×480	BQ Mall	1.6	0.2	69.97
	Basket Ball Drill	1.5	0.5	64.05
	Race Horse	1.9	0.4	64.30
	Party Scene	1.5	0.8	66.92
Class E 416×240	BQ Square	1.6	0.7	71.72
	Basket Ball Pass	1.1	0.6	74.65
	Race Horses	1.4	0.4	70.13
	Blowing Bubbles	1.8	0.8	71.46
Average		**1.61**	**0.52**	**67.44**

Table 6. Comparing our proposed coding unit decision algorithms with other state-of-the-art methods.

Algorithms	Bit Rate (BR) %	Encoding time(ΔT)%
Proposed support vector machine	1.61	67.44
[23]	1.7	53.00
[24]	0.88	57.21
[25]	1.45	59.64

5. EXPERIMENTAL RESULTS

The proposed ML-based approach was evaluated on test sequences of varied resolutions, and the results are outlined in Table 5. The approach achieved a 67.44 % decrease in average Encoding Time (ΔT) compared to the original HEVC, although there was a slight increase in Bit Rate (BR). To train the machine learning algorithm, the luminance values of the images were determined, and features such as RMSE, DC, SCCD, SD, and QP were defined. The training dataset consisted of a substantial number of images with associated luminance values. The algorithm was trained using 4000 blocks of size 64×64, 16000 blocks of size 32×32, and 640,000 blocks of size 16×16. The SVM algorithm was used to calculate depth values for each YUV sequence. These depth values for the best CUs were generated and incorporated into the HM 15.0 master to complete the HEVC procedure and achieve the desired results. Different classes were utilized in the algorithm. The HM 15.0 reference software was employed to implement additional quantization parameters (22, 27, 32, and 37).The test videos, featuring different resolutions,ran on a Windows 10 computer powered by an Intel(R) Xeon® W-2133 CPU running at 3.60 GHz, with 3600 MHz, 6 Cores, and 12 Logical Processors, using an All-Intra-Main configuration.

T_{prop} indicates the proposed encoding time,the encoding time of orginal is denoted by $T_{HM15.0}$, and ΔT is the percentage of saved encoding time. In Table 6, the suggested approach is contrasted with the other fast coding unit partition methods in terms of ΔT, BP, and BR. The proposed method lowered complexity with a slight increase in bit rate while reducing the average encoding time by 67.44 % compared to the several fast-coding unit decision algorithms.

6. CONCLUSION

This paper reduced the fast-coding unit partition complexity by using different image characteristics. Multiple features like RMSE, SD, SCCD, DC, and quantization parameters are calculated to extract the complete texture image information. Using homogeneous and non-homogeneous classification combined with brute force search will reduce computational complexity and improve classification accuracy. The proposed SVM model achieved the highest accuracy for splitting a sample using Directional Complexity and Standard Deviation. For all yuv sequences, the performance of the optimal depths is calculated, and these prediction depths are included in HM 15.0. The suggested machine learning approach for estimating the depths utilized less encoding time than the other coding unit partition methodologies, by roughly 67.44 %, with a slight increase in a bit rate of 1.61 %.

6.1 HEVC Application in the Defence Domain

Traditional HEVC reduces the bit rate by 50 % compared to H.264, whereas our suggested method reduces the encoding time by more than 50 % compared to traditional HEVC. Consequently, our proposed method is more appropriate for encoding high-quality video sequences, such as 4K and 8K resolutions. The high-resolution Image/Video coding is helpful for medical imaging, Satellite Imaging, and surveillance at borders with clearer images/Video. This feature is more useful for capturing and retrieving small objects from Indian border attackers. It is also useful for drone photography/videography.

REFERENCES

1. Wiegand, T.; Sullivan, G.J.; Bjøntegaard, G. & Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**(7),

560–576.
doi: 10.1109/TCSVT.2003.815165.

2. Sullivan, G.J.; Ohm, J.R.; Han, W.J. & Wiegand, T. Overview of the High-Efficiency Video Coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**(12),1649–1668.
doi: 10.1109/TCSVT.2012.2221191.
3. Bossen, F.; Bross, B.; Suhring, K. & Flynn, D. HEVC complexity and implementation analysis. *IEEE Transact. Circuits Syst. Video Technol*. 2012, **22**(12), 1685–1696.
doi: 10.1109/TCSVT.2012.2221255.
4. Lim, K.; Lee, J.; Kim, S. & Lee, S. Fast PU skip and split termination algorithm for HEVC intra prediction. *IEEE Trans. Circuits Syst. Video Technol.*, 2015, **25**(8), 1335–1346.
doi: 10.1109/TCSVT.2014.2380194.
5. Chao Tsai, An. & Bharanitharan, K. Multi-directional mode reduction algorithm for intra prediction in video coding. *IETE Technical Rev.*, 2016, **34**(1), 102-109.
doi:10.1080/02564602.2016.1151386.
6. Zhang, T.; Sun, M.T.; Zhao, D. & Gao, W. Fast intra-mode and CU size decision for HEVC. *IEEE Trans. Circuits Syst. Video Technol.,* 2015, **27**(8), 1714–1726.
doi: 10.1109/TCSVT.2016.2556518.
7. Lakshmi, P. & Aparna, P. A mixed parallel and pipelined efficient architecture for intra prediction schemein HEVC. *IETE Technical Rev.*, 2022, **39**(2), 244-256(2022).
doi:10.1080/02564602.2020.1841686.
8. Chen, Z.Y. & Chang, P.C. Rough mode cost-based fast intra coding for high-efficiency video coding. *J. Vis. Commun. Image Represent.,* 2017, **43**, 77–88.
doi: 10.1016/j.jvcir.2016.12.007.
9. Lim, K.; Lee, J. & Kim, S. Fast PU skip and split termination algorithm for HEVC intra prediction. *IEEE Trans. Circuits Syst. Video Technol.,* 2015, **25**(8), 1335–1346.
doi: 10.1109/TCSVT.2014.2380194.
10. Zhang, H. & Ma, Z. Fast intra-mode decision for high-efficiency video coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.,* 2014, **24**(4), 660–668.
doi: 10.1109/TCSVT.2013.2290578.
11. Yang, M. & Christos, C. Fast intra-encoding decisions for high-efficiency video coding standard. *J. Real-Time Image Process.,* 2017, **13**, 797–806.
doi: 10.1007/s11554-014-0445-7.
12. Huangyuan, Q.; Song, L.; Luo, Z.; Wang, X. & Zhao, Y. Performance evaluation of H.265/MPEG-HEVC encoders for 4K video sequences. Signal and Information Processing. In Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific. pp.1-8.
doi:10.1109/APSIPA.2014.7041782.
13. Lee, D. & Jeong, J. Fast CU size decision algorithm using machine learning for HEVC intra coding. *Signal Process. Image Commun.*, 2018, **62**, 33–41.
doi:10.1016/j.image.2017.12.005.
14. Bai, C.& Yuan, C. Fast coding tree unit decision for HEVC intra coding. *In* 2013 IEEE International Conference on Consumer Electronics - China, April 2013, pp. 28–31.
doi:10.1109/ICCE-CHINA.2013.6780861.
15. Li, T.; Xu, M.& Deng, X. A deep convolutional neural network approach for complexity reduction on intra-mode HEVC. *In* 2017 IEEE International Conference on Multimedia and Expo (ICME), July 2017, pp. 1255–1260.
doi: 10.1109/ICME.2017.8019316.
16. Imen, W.; Amna, M. & Fatma, B. Fast HEVC intra-CU decision partition algorithm with modified LeNet-5 and AlexNet. SIViP, 2022, **16**, 1811-1819.
doi: 10.1007/s11760-022-02139-w.
17. Guanwen, Zhang.; Liyuan, Xiong.; Xiaocong, Lian. & Wei, Zhou1. A CNN-based coding unit partition in HEVC for video processing. *In* Proceedings of the 2019 IEEE International Conference on Real-time Computing and Robotics.August 4-9, 2019, pp. 273–276.
doi: 10.1109/RCAR47638.2019.9043972.
18. Kim, K. & Ro, W.W. Fast CU depth decision for HEVC using neural networks. *IEEE Transact. Circuits Syst. Video Technol.*, 2019, **29**(5), 1462-1473.
doi:10.1109/TCSVT.2018.2839113.
19. Wang, Z. & Li, F. Convolutional neural network based low complexity HEVC intra encoder. *Multimedia Tools Appl.,* 2021, **80**, 2441–2460.
doi:10.1007/s11042-020-09231-8.
20. Bouaafia, S.; Khemiri, R.; Maraoui, A. & Fatma Elzahra, S. CNN-LSTM Learning approach-based complexity reduction for high-efficiency video coding standard. *Scientific Programming.,* 2021.
doi: 10.1155/2021/6628041.
21. Bouaafia, S.; Khemiri, R.& Sayadi, F.E. Fast CU partition-based machine learningapproach for reducing HEVC complexity. *J. Real-Time Image Proc.*, 2020, **17**, 185–196.
doi: 10.1007/s11554-019-00936-0.
22. Grellert, M; Zatt, B.; Bampi, S.& da Silva Cruz, L.A. Fast coding unit partition decision for HEVC using support vector machines. *IEEE Transact. Circuits Syst. Video Technol.,* 2019, **29**(6),1741-1753.
doi:10.1109/TCSVT.2018.2849941.
23. Zhang, M.; Zhai, X. & Liu, Z. Fast and adaptive mode decision and CU partition early termination algorithm for intra-prediction in HEVC. *J. Image Video Proc.*, 2017, **86**.
doi: 10.1186/s13640-017-0237-7.
24. Song, Y.; Zeng, Y. & Li, X. Fast CU size decision and mode decision algorithm for intra prediction in HEVC. *Multimedia Tools Appl.,* 2017, **76**, 2001–2017.
doi: 10.1007/s11042-015-3155-7.
25. Zhu, W.; Yi, Y. & Zhang, H. Fast mode decision algorithm for HEVC intra coding based on texture partition and direction. *J. Real-Time Image Proc.,* 2020, **17**, 275–292.
doi:10.1007/s11554-018-0766-z.

CONTRIBUTORS

Mr Pattimi Hari obtained his MTech degree in Electronics and Communication Engineering from JNTU Kakinada, Andhra Pradesh, India.

In the current study, he developed the approach, conducted simulations, explained the findings, and wrote the manuscript.

Dr Kota Naga Srinivasarao Batta obtained his PhD degree from IIT Kharagpur, India. He is working as an Assistant Professor in the Department of ECE, National Institute of Technology, Warangal. His research interests include: VLSI architectures for image and video compression and embedded systems design.

In the current study, he has given guidance for how to develop the models in video coding. He has revised the manuscript also.