

RESEARCH

Open Access



# Fast CU size decision and intra-prediction mode decision method for H.266/VVC

Mingying Li<sup>1</sup>, Zhiying Wang<sup>2\*</sup>  and Qiuwen Zhang<sup>2</sup>

\*Correspondence:  
332107040640@email.zzuli.  
edu.cn

<sup>1</sup> Henan Finance University,  
Zhengzhou 450046, China

<sup>2</sup> College of Computer  
and Communication  
Engineering, Zhengzhou  
University of Light Industry,  
No. 22 North Longzihu Road,  
Zhengzhou 450002, China

## Abstract

H.266/Versatile Video Coding (VVC) is the most recent video coding standard developed by the Joint Video Experts Team (JVET). The quad-tree with nested multi-type tree (QTMT) architecture that improves the compression performance of H.266/VVC is introduced. Moreover, H.266/VVC contains a greater number of intra-prediction modes than H.265/High Efficiency Video Coding (HEVC), totalling 67. However, these lead to extremely the coding computational complexity. To cope with the above issues, a fast intra-coding unit (CU) size decision method and a fast intra-prediction mode decision method are proposed in this paper. Specifically, the trained Support Vector Machine (SVM) classifier models are utilized for determining CU partition mode in a fast CU size decision scheme. Furthermore, the quantity of intra-prediction modes added to the RDO mode set decreases in a fast intra-prediction mode decision scheme based on the improved search step. Simulation results illustrate that the proposed overall algorithm can decrease 55.24% encoding runtime with negligible BDBR.

**Keywords:** CU size decision, Intra-prediction modes, SVM, H.266/VVC

## 1 Introduction

As the video domain continues to advance, the ultra-high definition (UHD) video has become more and more popular because it can bring people a better experience, the video coding technology is facing higher demands [1]. The previous generation video coding standard falls short of satisfying the compression demands of the future market [2], while the increasing density of data poses significant challenges in terms of bandwidth and storage capacity. Consequently, the development of H.266/VVC is urgent [3], which can suffice the requirements for clarity, fluency, and real-time video. In addition, the bit-depth of H.266/VVC is 10-bit and is formulated for 4 K/8 K video. The maximum CU size of the encoder becomes 128 resulting from this reason. For instance, even if the input is an 8-bit sequence in the encoding process, it is converted to 10-bit for processing. The H.266/VVC not only inherits some technologies of H.265/HEVC but also expands upon the original partition structure [4]. The compression efficiency in H.266/VVC has been efficiently enhanced due to the incorporation of numerous advanced coding techniques, but causes an extreme rise in the coding computational intricacy and reduces the encoding speed of H.266/VVC [5]. The computational intricacy of VTM is

19 times without any exaggeration compared with HM under the “All Intra” configuration [6]. Consequently, effectively mitigating the encoding computational burden and efficiently compressing vast quantities of data have emerged as significant challenges in the practical implementation of H.266/VVC.

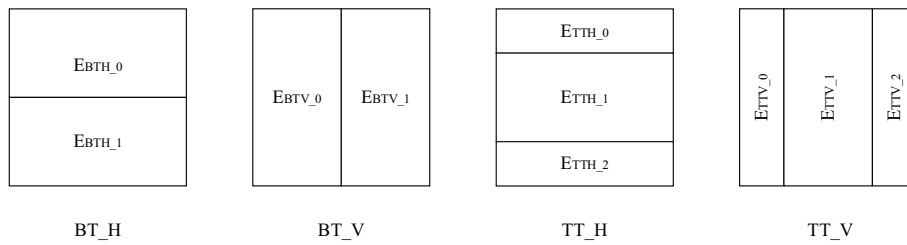
The researchers have proposed many different methods for H.265/HEVC to optimize the intra-prediction structure and enhance the coding efficiency. In Ref. [7], a rate-distortion-based CU partition network is introduced for division decisions guided by statistical characteristics. In Ref. [8] a convolutional neural network that relies on multi-scale information fusion is introduced to provide a swift and precise prediction of CU partition for both intra-and inter-modes. In Ref. [9], an HEVC intra-coding chroma image enhancement convolutional neural network guided by luminance is introduced, demonstrating the capability to generate high-quality chroma images. Fast intra-mode selection is performed by mapping the intra-modes of HEVC to fireflies in the firefly algorithm in Ref. [10]. Reference [11] developed algorithms based on LeNet-5 and AlexNet to optimize the intra-prediction CU partition module in HEVC to reduce a large amount of computing time. In Ref. [12], spatial correlation with neighboring CTUs is harnessed to constrain the rate-distortion optimization for individual CTUs, effectively decreasing the range of candidate CU sizes to be contemplated for individual CTUs. Reference [13] proposed an efficient algorithm that minimizes computational complexity by skipping unnecessary patterns and maintaining coding efficiency. An algorithm was proposed in Ref. [14] to reduce the complexity of intra-prediction in HEVC by training three CNN models to predict the best CU partition of the QT step. Reference [15] proposed a lightweight model for intra-mode early decision-making that can dynamically adjust the best option. Reference [16] proposes to combine deep learning and parallel computing technologies and apply them in HEVC intra-mode to substantially decrease the intricacy of the HEVC encoding engine while preserving RD performance. In Ref. [17], the reinforcement learning framework employs the Frank Wolfe strategy to address the CTU bit allocation challenge in HEVC for region-of-interest intra-coding, which can effectively transform the CTU-level bit allocation problem into an action-constrained reinforcement learning problem. All these fast algorithms effectively reduce the H.265/HEVC coding complexity to some extent. Since the partition architecture of H.266/VVC is very different from that of H.265/HEVC, the newly added coding technologies cause a surge complexity in intra-prediction. Therefore, if the previous algorithms for H.265/HEVC are applied to H.266/VVC, which can inevitably cause a higher computational complexity.

Recently, some fast CU partitioning decision applied to H.266/VVC has appeared for decreasing the coding burden. Reference [18] proposed a low-complexity intra-coding decision, which includes a fast Bayesian-based CU segmentation decision and a fast L-BFGS-based intra-mode decision. A Random Forest-based algorithm is used in Ref. [19] to train Random Forest classifiers for different classes of CUs separately to directly predict the optimal division pattern. A simple early decision tool is proposed in Ref. [20] by identifying features in the encoding context of H.266/VVC, which can usefully alleviate the coding burden. A fast partition method in light of variance and gradient is exploited in Ref. [21] for tackling the asymmetric splitting problem brought by QTMT partition structure, thus ensuring both coding performance and compression quality. In

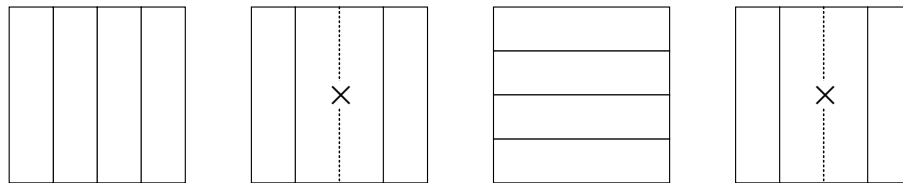
Ref. [22], a fast intra-partitioning algorithm was proposed based on variance and Sobel operators, which can effectively solve the new asymmetric partitioning structure problem in VVC. Reference [23] applies the LGBM classifier to the intra-partitioning of VVC to bypass the exhaustive RDO process and skip the less likely CU partitioning types. Reference [24] proposed an algorithm that uses the difference obtained from the gradient and content of sub-blocks to eliminate redundant horizontal and vertical partitioning patterns. In Ref. [25], a multi-stage CNN equipped with an early exit mechanism is suggested to forecast the CU partitioning mode within VVC, significantly expediting the intra-mode encoding process. In Ref. [26], different decision models are established for different partition modes by obtaining the distortion of CU, and proposed an adjustable CU decision-making model to achieve complexity reduction. Reference [27] utilized human vision models with significant differences to identify visually discernible pixels within CUs, and employs a random forest model to forecast MTT partition while eliminating unnecessary partitions. To accurately forecast the QTMT structure, Ref. [28] presents a hierarchical grid fully CNN, which necessitates only a single inference to retrieve comprehensive partition details of the current CU and its sub-CUs, enabling the direct prediction of the specific hierarchical partitioning structure. In Ref. [29], the FSVM model and DAG-SVM model are harnessed to forecast both the necessity of CU partitioning and the specific partition type, which can effectively circumvents superfluous RDO processes within VVC intra-prediction. An adaptive CU partitioning decision method utilizes the pooling-variable CNN for various CU shape which is exploited in Ref. [30] to decrease the encoding complexity. Inspired by the above methods, this paper presents novel algorithms for fast intra-CU size decision and efficient intra-prediction in H.266/VVC, whose objective is to efficiently decrease the coding computational burden.

In previous some works, many researchers are committed to studying CU splitting and mode decision. Many fast methods are introduced for decreasing computational intricacy. To resolve the asymmetric splitting question, we design a fast intra-CU size and fast intra-prediction mode decision method to reduce the encoding computational burden. The fast CU sizing decision scheme first selects effective features to distinguish CU partition mode. There are three situations: when CU sizes are  $128 \times 128$  and  $64 \times 64$ , the effective features include the texture variance of a whole CU, the expected value of partition, variance of partition, and quantization parameter (QP); when CU sizes are  $32 \times 16$ ,  $16 \times 32$ ,  $16 \times 16$ ,  $8 \times 16$ , and  $16 \times 8$ , the effective features include the entropy variance difference, the difference between the texture contrast and Haar. Subsequently, the SVM classifier models corresponding to CU size are trained on-line using these characteristics. Eventually, the trained SVM classifier models are utilized for determining CU partition mode; when CU size is  $32 \times 32$ , the variances of the variance of the five splitting modes are calculated separately and each value corresponds to one splitting mode. The best splitting of the CU is determined by selecting the maximum value of the splitting mode. Furthermore, a fast intra-prediction mode decision scheme is devised, where the quantity of intra-prediction modes added to the RDO mode set decreases through improved search step. Overall, the proposed algorithm cannot only significantly decrease the coding complexity but also ensure video quality at the same time.

The rest of this paper is structured as follows. The motivation and analysis of the proposed scheme are shown in Sect. 2. The main idea of the fast CU size decision method



**Fig. 1** The partition types of MTT architecture



**Fig. 2** Redundant splitting modes of BT and TT

and fast intra-prediction mode decision scheme is illustrated in Sect. 3. Section 4 illustrates the experimental results of the proposed approach and provides a comparison with the other fast approaches. Section 5 offers the concluding remarks for this paper.

## 2 Motivation and analysis

Only QT partition architecture is allowed in H.265/HEVC, while the partition structure of H.266/VVC allows the asymmetric partitioning. In theory, the CU size in H.266/VVC can be configured with any combination of 128, 64, 32, 16, 8, or 4. In addition, the smaller length (2 or 1) exists resulting from the introduction of intra-sub-partitions (ISP) technique [31]. Consequently, the multi-type tree (MTT) partition structure can obtain more efficient coding performance but greatly rises the computational intricacy. The CTUs are first divided into four leaf nodes in H.266/VVC. Subsequently, the leaf nodes undergo further partition through the employment of the MTT partition architecture. If the CU sizes are  $128 \times 128$  and  $64 \times 64$ , which is only split by QT partition architecture since the size exceeds the maximum allowable MTT size. Otherwise, it is further split through MTT partition architecture. The four partition types in the MTT structure are shown in Fig. 1, which include horizontal BT (BT\_H), vertical BT (BT\_V), horizontal TT (TT\_H), and vertical TT (TT\_V).

Since the different splitting modes could potentially result in the same CU size, some redundant splitting modes are disallowed in H.266/VVC. The redundancy partitioning pattern in the MTT structure is shown in Fig. 2. It can be seen that two consecutive levels BT splitting in one direction could have the same CU size as TT splitting followed by BT splitting in the central partition. Thus, the syntax prevents BT splitting (in the given direction) for the central partition of TT splitting. These restrictions apply to CUs in all pictures.

A series of experiments for H.266/VVC are implemented on the VTM 7.0 to dissect the complexity of MTT splitting architecture under the QP values (22, 27, 32, 37). The

average time-consuming of MTT architecture is about 95% in various sequences, that is, the MTT architecture in H.266/VVC is the most time-consuming part. Specifically, the proportions of different CU partition results are analyzed. For sequences belonging to the smooth area, the CUs might not be further split. For sequences belonging to a complex area, the CUs are likely to be further partitioned into small CU by MTT structure. Thus, it is noticed that the texture traits have an intense connection with the CU partition. The texture traits of sequences are utilized to early terminate the coding procedure of CUs and reduce the coding computational burden. According to the aforementioned analysis, it is an important task to seek out a method to lessen the computational burden of CU partitioning.

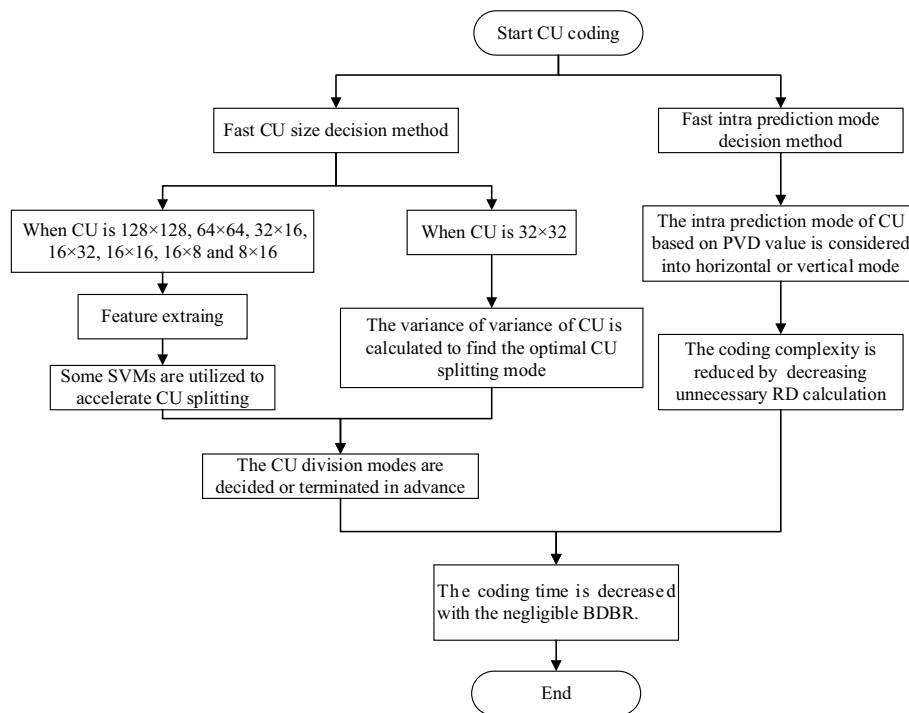
Furthermore, the quantity of intra-prediction patterns in H.266/VVC is augmented from 35 to 67 for better adjusting to the various types of local features in natural videos. Although the 67 intra-modes offer outstanding compression gains than H.265/HEVC, selecting the optimal intra-mode leads to the high coding computational intricacy. Thus, it summarizes that the encoding runtime of the RDO procedure surpasses the half of the total encoding runtime on the basis of the statistical data. The mode decision procedure of VTM finds the best prediction pattern with the least RD cost through an exhaustive search, and the RD cost is calculated as

$$J_{\text{SATD}} = D_{\text{Had}} + \lambda_{\text{MODE}} \times R_{\text{MODE}} \quad (1)$$

where  $D_{\text{Had}}$  is the Hadamard transform of the CU residual.  $R_{\text{MODE}}$  represents the number of bits generated by the transformation.  $\lambda$  refers to the Lagrangian multiplier. This method results in an extremely computational burden. Consequently, to further decrease the encoding time, a fast intra-prediction mode decision algorithm is also proposed.

### 3 Proposed algorithm

To cope with the asymmetric partition problem brought by the QTMT splitting architecture, a fast approach for intra-CU size decision and an efficient scheme for intra-prediction mode decision scheme are designed to reduce the coding computational intricacy. Figure 3 shows the overall flowchart of the proposed algorithm. In the fast CU size decision method, the effective features are firstly chosen to determine the CU splitting mode. Subsequently, the SVM classifier models corresponding to CU sizes are trained on-line using these characteristics. Eventually, the SVM classifier models are utilized for determining CU partition mode. Taking into account the different situations of different video sequences, the SVM classifier models are trained on-line and updated regularly, with each period consisting of 80 frames. Specifically, the SVM classifier models are trained using the first frame, and the subsequent 79 frames are utilized for prediction. When CU size is equal to  $32 \times 32$ , we will compute the variance of the five partition modes separately. Use the partition mode corresponding to the maximum value as the optimal partition mode of the CU. Furthermore, the proposed fast intra-prediction mode decision method is designed to decrease the complexity by improving the search step and reducing the number of predictive modes added to the RDO pattern set.



**Fig. 3** The flowchart of the proposed overall method

### 3.1 Feature extraction and analysis

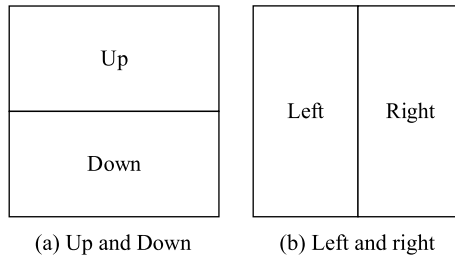
The spatial feature used in H.265/HEVC is not directly applicable to H.266/VVC. If the CU sizes are  $128 \times 128$  and  $64 \times 64$ , only the QT partition structure is allowed to split CU since the size exceeds the maximum allowable MTT size. Furthermore, CUs generally select the larger size when the sequences are smooth and CUs familiarly choose the smaller size when the sequences are complex. Accordingly, the texture features and CU size have a strong relationship, where the variance is employed as a metric to the texture complexity that is presented as

$$\xi = \sqrt{\frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=1}^{H-1} (x_{ij} - \bar{x})^2} \quad (2)$$

$$\bar{x} = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=1}^{H-1} x_{ij} \quad (3)$$

where  $W$  and  $H$  are the width and height of the CU,  $\bar{x}$  is the average values of pixel, and  $\xi$  denotes the texture variance of a whole CU.

The up and down or the left and right of the CU should be very similar when CU splitting is early terminated. Figure 4 shows the up, down, left, and right position in a CU. Therefore, the expected values and variance absolute difference in the up and down of a CU forecast whether the two parts exhibit differences or not. The same applies to both the left and



**Fig. 4** The up, down, left, and right positions in a CU

right parts of the CU. The absolute difference between the expected values and variance is expressed as

$$e_v = |e_u - e_d|, \xi_v = |\xi_u - \xi_d| \tag{4}$$

$$e_h = |e_l - e_r|, \xi_h = |\xi_l - \xi_r| \tag{5}$$

where  $e_v$  and  $e_h$  refer to the vertical and horizontal absolute difference of the expected value, respectively.  $\xi_v$  and  $\xi_h$  refer to the vertical and horizontal absolute difference of variance, respectively.  $e_u, e_d, e_l,$  and  $e_r$  are the expected values of up, down, left and right, respectively.  $\xi_u, \xi_d, \xi_l,$  and  $\xi_r$  are the variance values of up, down, left and right, respectively. When CU splitting is early terminated,  $e_v, e_h, \xi_v,$  and  $\xi_h$  should be small. Thus, the sum of the absolute difference of the expected values and variance is defined as the expected value and variance of partition, respectively:

$$e_s = e_v + e_h \tag{6}$$

$$\xi_s = \xi_v + \xi_h \tag{7}$$

where  $e_s$  and  $\xi_s$  denote the sum of the absolute difference of the expected values and variance. Moreover, QP has also an effect on the CU size decision. CU generally chooses the large size when QP setting is large. In contrast, CU is likely to select the small size when QP is small. Building upon the aforementioned analysis, the texture variance of a whole CU, the expected value of partition, variance of partition, and QP are considered as the feature vector when the CU sizes are  $128 \times 128$  and  $64 \times 64$ .

When CU sizes are  $32 \times 16, 16 \times 32, 16 \times 16, 8 \times 16,$  and  $16 \times 8,$  the improved features based on entropy and texture contrast are extracted, where entropy and texture contrast denote the information quantity. The partition modes of MTT architecture as shown in Fig. 1. The improved features are denoted as  $\Delta E_{BT}, \Delta E_{TT}, \Delta T_{BT}$  and  $\Delta T_{TT},$  which are defined as

$$\Delta E_{BT} = |E_{BTH_0} - E_{BTH_1}| - |E_{BTV_0} - E_{BTV_1}| \tag{8}$$

$$\Delta E_{TT} = |E_{TTH_0} - E_{TTH_1}| + |E_{TTV_1} - E_{BTV_2}| - |E_{TTV_0} - E_{TTV_1}| - |E_{TTV_1} - E_{TTV_2}| \tag{9}$$

where  $\Delta E_{BT}$  is the entropy variance difference in BT\_H and BT\_V,  $|E_{BTH\_0} - E_{BTH\_1}|$  and  $|E_{BTV\_0} - E_{BTV\_1}|$  denote the entropy variance of BT\_H and BT\_V, respectively. Similarly,  $\Delta E_{TT}$  is the entropy variance difference in TT\_H and TT\_V.  $|E_{TTH\_0} - E_{TTH\_1}| + |E_{TTV\_1} - E_{BTV\_2}|$  is the entropy variance of TT\_H and  $|E_{TTV\_0} - E_{TTV\_1}| - |E_{TTV\_1} - E_{TTV\_2}|$  represents the entropy variance of TT\_V. Moreover,  $E_K$  represents the entropy of sub-CUs, which is presented as

$$E_K = - \sum_{i=0}^{255} p(i) \log p(i) \quad (10)$$

where  $p(i)$  denotes the probability of the  $i$ -th gray scale value.  $\Delta T_{BT}$  and  $\Delta T_{TT}$  are defined as

$$\Delta T_{BT} = |T_{BTH\_0} - T_{BTH\_1}| - |T_{BTV\_0} - T_{BTV\_1}| \quad (11)$$

$$\Delta T_{TT} = |T_{TTH\_0} - E_{TTH\_1}| + |T_{TTV\_1} - T_{BTV\_2}| - |T_{TTV\_0} - T_{TTV\_1}| - |T_{TTV\_1} - T_{TTV\_2}| \quad (12)$$

where  $\Delta T_{BT}$  is the difference between the texture contrast of BT\_H and BT\_V.  $|E_{BTH\_0} - E_{BTH\_1}|$  and  $|E_{BTV\_0} - E_{BTV\_1}|$  denote the texture contrast of BT\_H and BT\_V, respectively. Similarly,  $\Delta T_{TT}$  is the difference between the texture contrast of TT\_H and TT\_V.  $|T_{TTH\_0} - E_{TTH\_1}| + |T_{TTV\_1} - T_{BTV\_2}|$  and  $|T_{TTV\_0} - T_{TTV\_1}| - |T_{TTV\_1} - T_{TTV\_2}|$  represent the texture contrast of TT\_H and TT\_V, respectively. Furthermore,  $T_K$  is the texture of sub-CUs, which is calculated as

$$T_K = \sum_{i=1}^W \sum_{j=1}^H (i-j)^2 f(i,j) \quad (13)$$

where  $W$  and  $H$  denote the width and height of sub-CU, respectively.  $f(i,j)$  represents the pixel value at position  $(i,j)$ .

Furthermore, Haar reflects the gray level changes of the image and CU partitioning patterns can be accurately predicted, which is expressed by

$$D = \frac{4}{W \times H} \sum_{i=1}^{W/2} \sum_{j=1}^{H/2} \left( \left| \begin{array}{c} f(2 \times i, 2 \times j) - f(2 \times i, 2 \times j + 1) + f(2 \times i + 1, 2 \times j) - f(2 \times i + 1, 2 \times j + 1) \\ f(2 \times i, 2 \times j) + f(2 \times i, 2 \times j + 1) - f(2 \times i + 1, 2 \times j) - f(2 \times i + 1, 2 \times j + 1) \end{array} \right| \right) \quad (14)$$

where  $|f(2 \times i, 2 \times j) - f(2 \times i, 2 \times j + 1) + f(2 \times i + 1, 2 \times j) - f(2 \times i + 1, 2 \times j + 1)|$  and  $|f(2 \times i, 2 \times j) + f(2 \times i, 2 \times j + 1) - f(2 \times i + 1, 2 \times j) - f(2 \times i + 1, 2 \times j + 1)|$  represent the horizontal and vertical coefficient in Haar wavelet transform, respectively. According to the aforementioned analysis, the effective characteristics including the entropy variance difference, the difference between the texture contrast and Haar are considered as the feature vector when the CU sizes are  $32 \times 16$ ,  $16 \times 32$ ,  $16 \times 16$ ,  $8 \times 16$ , and  $16 \times 8$ .

The F-score method is utilized to determine the most effective features for classification from a multitude of options. Moreover, the F-score calculation is uncomplicated, which is calculated as

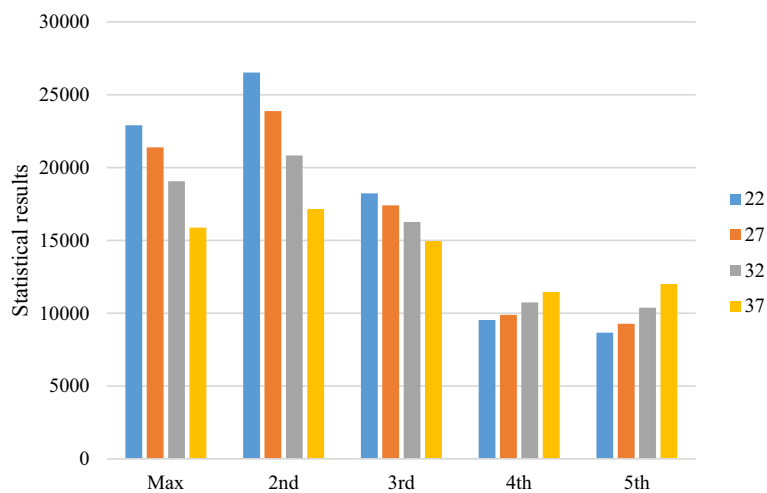


$$F_i = \frac{(\bar{x}_i^{(pos)} - \bar{x}_i)^2 + (\bar{x}_i^{(neg)} - \bar{x}_i)^2}{\frac{1}{n_{pos}-1} \sum_{l=1}^{n_{pos}} (x_{l,i}^{(pos)} - \bar{x}_i^{(pos)})^2 + \frac{1}{n_{neg}-1} \sum_{l=1}^{n_{neg}} (x_{l,i}^{(neg)} - \bar{x}_i^{(neg)})^2} \quad (15)$$

where  $n_{pos}$  and  $n_{neg}$  represent the amount of positive and negative samples, respectively.  $\bar{x}_i$  refers to the average value of the  $i$ -th feature vector.  $\bar{x}_i^{(pos)}$  and  $\bar{x}_i^{(neg)}$  refer to the average value of the  $i$ -th feature vector in the positive and negative sample sets, respectively.  $\bar{x}_{l,i}^{(pos)}$  and  $\bar{x}_{l,i}^{(neg)}$  represent the eigenvalues of the  $i$ -th feature vector of the  $l$ -th positive and negative sample points, respectively. In the proposed fast CU size decision method, when the CU sizes are  $128 \times 128$  and  $64 \times 64$ , the split and non-split are set as the positive and negative; when CU sizes are  $32 \times 16$ ,  $16 \times 32$ ,  $16 \times 16$ ,  $8 \times 16$ , and  $16 \times 8$ , the horizontal and vertical splitting are set as the positive and negative. Table 1 displays the F-score values of the various features. It is observed that five sequences with different resolutions are selected for calculating the F-score value that is its value of SVM classifier models. Moreover, it is noticed that the discriminativeness of a feature increases with the increase of F-score value.

**Table 1** The F-score values of features

Features	Sequences				
	RaceHorses	BQMall	Johnny	Cactus	Campfire
$\xi$	0.121	0.098	0.087	0.098	0.131
$e_s$	0.098	0.127	0.096	0.126	0.096
$\xi_s$	0.102	0.124	0.099	0.089	0.101
QP	0.085	0.091	0.088	0.097	0.102
$\Delta E_{BT}$	0.053	0.095	0.075	0.095	0.061
$\Delta E_{TT}$	0.055	0.101	0.079	0.093	0.069
$\Delta T_{BT}$	0.079	0.124	0.125	0.123	0.055
$\Delta T_{TT}$	0.055	0.112	0.115	0.058	0.061
$D$	0.055	0.042	0.099	0.093	0.079



**Fig. 5** The statistical results of under different QPs

In addition, the different textures are partitioned into different size sub-CUs for realizing a better forecasting property. Part of CUs has been resolved for partition, while QT or MTT structure is used to split when CU size is  $32 \times 32$ . According to the statistical analysis based on the 100 images in the DIV2K data set, the variance of the variance of each sub-CUs is obtained. Figure 5 illustrates the statistical results under different QPs. It is noticed from Fig. 5 that the maximum variance of variance corresponding to splitting mode seems to be the probable mode.

We can reasonably conjecture the comparatively large variance of variance in five splitting modes as the best mode. Therefore, the variance of the variance of each sub-CU in five splitting modes is devised for further reducing the coding computational burden, where the maximum value corresponding to partition mode is considered as the best partition mode. The variance of the variance of sub-CU is expressed as

$$V_{QT} = \frac{1}{4} \left( \sum_{n=1}^4 \left( \frac{1}{W_n \times H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (X(i,j) - \xi_n)^2 - \xi_{QT} \right) \right)^2 \quad (16)$$

$$V_{BT\_H} = \frac{1}{2} \left( \sum_{n=1}^2 \left( \frac{1}{W_n \times H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (X(i,j) - \xi_n)^2 - \xi_{BT} \right) \right)^2 \quad (17)$$

$$V_{BT\_V} = \frac{1}{2} \left( \sum_{n=1}^2 \left( \frac{1}{W_n \times H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (X(i,j) - \xi_n)^2 - \xi_{BV} \right) \right)^2 \quad (18)$$

$$V_{TT\_H} = \frac{1}{3} \left( \sum_{n=1}^3 \left( \frac{1}{W_n \times H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (X(i,j) - \xi_n)^2 - \xi_{TH} \right) \right)^2 \quad (19)$$

$$V_{TT\_V} = \frac{1}{3} \left( \sum_{n=1}^3 \left( \frac{1}{W_n \times H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (X(i,j) - \xi_n)^2 - \xi_{TV} \right) \right)^2 \quad (20)$$

$$V_M = \max (V_{QT}, V_{BT\_H}, V_{BT\_V}, V_{TT\_H}, V_{TT\_V}) \quad (21)$$

where  $V_{QT}$ ,  $V_{BT\_V}$ ,  $V_{BT\_H}$ ,  $V_{TT\_V}$  and  $V_{TT\_H}$  denote the variance of the variance of each sub-CU in QT, BT\_V, BT\_H, TT\_V, and TT\_H.  $W_n$ ,  $H_n$ , and  $\xi_n$  designate the width, height, and the mean value of pixels of the  $k$ -th sub-CU, respectively.  $\xi_{QT}$ ,  $\xi_{BV}$ ,  $\xi_{BH}$ ,  $\xi_{TV}$  and  $\xi_{TH}$  represent the average variance value of all sub-CUs in QT, BT\_V, BT\_H, TT\_V, and TT\_H.  $V_M$  is the maximum value in five partition modes.

### 3.2 Fast CU size decision method

The SVM classifier model can settle the two-class classification problem, which can seek out the best hyperplane in two classes. The optimal hyperplane leads to improved classification performance in feature space, where the training sample set is defined as

$$D = \left\{ (x_i, y_i), x_i \in R^N, y_i \in \{-1, +1\}, i = 1, 2, \dots, m \right\} \quad (22)$$

where  $y_i$  refers to the class label corresponding to the input feature vector  $x_i$ .  $R^N$  refers to the dimension of the characteristic vectors. The optimal hyperplane  $W \cdot x + b = 0$  can divide the training samples. Maximizing  $\frac{1}{\|W\|}$  and minimizing  $\frac{1}{2}\|W\|^2$  are equivalent, which is presented as

$$\begin{aligned} \min \Psi(W) &= \frac{1}{2}\|W\|^2 = \frac{1}{2}(W' \cdot W) \\ \text{s.t. } y_i[(W \cdot x_i + b)] &\geq 1, i = 1, 2, \dots, m \end{aligned} \tag{23}$$

where  $\frac{1}{\|W\|}$  denotes the classification interval. In addition, the perfect classification is nonexistent. For example, the samples of one class are misclassified as other classes. Thus, the optimal hyperplane can solve the misclassified classification by the error penalty for controlling the classification precision, and it is presented as

$$\begin{aligned} \min \tau(W, \mu) &= \frac{1}{2}\|W\|^2 + C \sum_{i=1}^m \mu_i \\ \text{s.t. } y_i(W^T x_i + a) &\geq 1 - \mu_i, \mu_i \geq 0, i = 1, 2, \dots, m \end{aligned} \tag{24}$$

where  $W$  denotes the normal vector obtained using original dual-relation.  $C$  is the penalty factor.  $\mu_i$  is slack variable.  $a$  denotes the bias variable. Further, the above optimization problem is settled by introducing the Lagrange multipliers:

$$\Gamma(W, \mu, a, \gamma, \eta) = \frac{1}{2}\|W\|^2 + C \sum_{i=1}^m \mu_i - \sum_{i=1}^m \gamma_i (y_i(W^T \Phi(x_i) + a) - 1 + \mu_i) - \sum_{i=1}^m \eta_i \mu_i \tag{25}$$

where  $\gamma_i$  and  $\eta_i$  denote Lagrange multiplier.  $\Phi(x_i)$  is a mapping. The solution of the hyperplane optimization question is given by the saddle point of the Lagrange function:

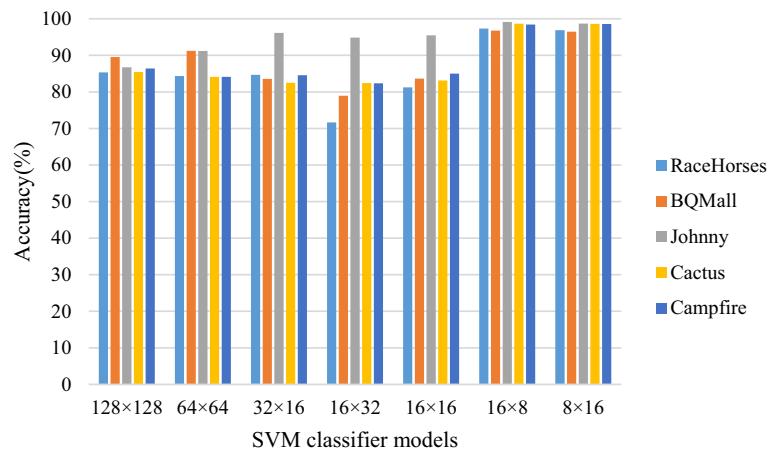
$$\begin{aligned} \max W(\gamma) &= \sum_{i=1}^m \gamma_i - \frac{1}{2} \sum_{i,j=1}^m \gamma_i \gamma_j y_i y_j \langle x_i, x_j \rangle \\ \text{s.t. } 0 \leq \gamma_i &\leq C \sum_{i=1}^m \gamma_i y_i, i = 1, 2, \dots, m \end{aligned} \tag{26}$$

Finally, the decision function is presented as

$$f(x) = \text{sign}\{W \cdot x + a\} = \text{sign}\left\{ \sum_{i=1}^m \gamma_i y_i \langle x, x_i \rangle + a \right\} \tag{28}$$

$$W = \sum_{i=1}^m \gamma_i y_i \Phi(x_i) \tag{27}$$

where  $\text{sign}\{\cdot\}$  is the signum function.  $\langle \cdot, \cdot \rangle$  refers to the scalar product. Specifically, when the CU sizes are  $128 \times 128$  and  $64 \times 64$ , the effective features including the texture variance of a whole CU, the expected value of partition, variance of partition, and QP are used for two SVM classifier models, in which  $+1$  and  $-1$  represent split and non-split, respectively. Similarly, when the CU sizes are  $32 \times 16$ ,  $16 \times 32$ ,  $16 \times 16$ ,  $8 \times 16$ , and  $16 \times 8$ , the corresponding effective features including the entropy variance difference,



**Fig. 6** The predicting precision of these SVM classifier models

the difference between the texture contrast and Haar are utilized for five SVM classifier models, where + 1 represents horizontal partition and – 1 denotes vertical partition. The SVM models are trained on-line and regularly updated since the different video sequences have different situations. Each period has 80 frames, the SVM classifier models are trained based on the data from the first frame, and the subsequent 79 frames are used for prediction. Figure 6 demonstrates the predicting precision of these SVM classifier models. It is noticed from Fig. 6 that the average precision of most SVM classifier models exceeds 80% and the average precision of classifiers for the small size of CUs surpasses 90%. Accordingly, it verifies that SVM classifier models are efficient for these CUs.

Moreover, the maximum variance of variance corresponding to splitting mode seems to be the probable mode when CU size is  $32 \times 32$  in light of the statistical analysis presented in Sect. 3.1. Consequently, the variance of the variance of each sub-CU in five splitting modes is calculated for further reducing the complexity of CU splitting, and the partitioning mode with the maximum value is regarded as the best. Figure 7 illustrates the flowchart of the proposed fast CU size decision scheme.

### 3.3 Fast intra-prediction mode decision algorithm

The intra-prediction modes in H.266/VVC have been increased from the original 35 of H.265 to 67, namely 65 directional prediction modes and DC, Planar. Significantly improving the accuracy of intra-prediction also brings a high computational burden to the RDO procedure. Therefore, a proposed fast intra-prediction mode decision scheme aims to minimize unnecessary RD calculations and reduce coding complexity. Figure 8 illustrates the flowchart of the proposed fast intra-prediction mode decision method.

Texture direction is an important feature in the intra-prediction mode in H.266/VVC. The utilization of texture direction allows for the removal of several unnecessary intra-prediction modes, which can simplify the intra-prediction procedure. The Planar or DC mode is often used for regions with flat texture and the directional modes are used for regions exhibiting intricate edges and fine details. The texture features are employed to ascertain whether the intra-prediction mode is the horizontal or vertical mode, where 2–34 modes

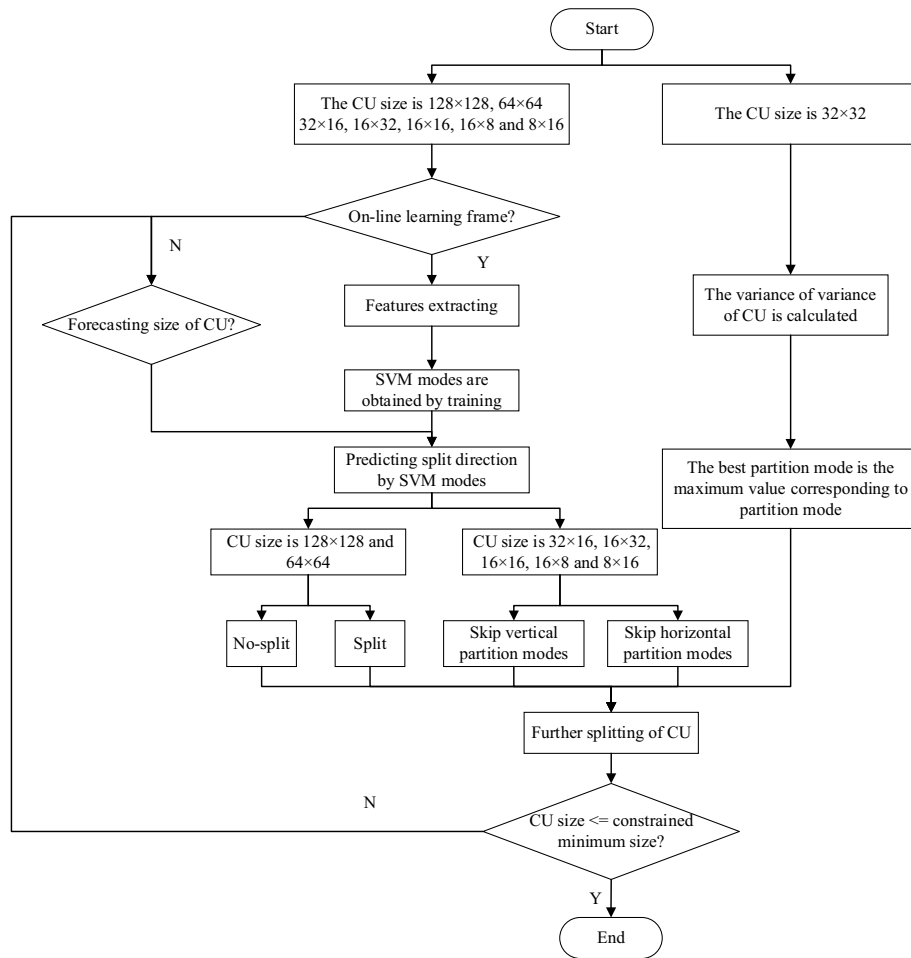


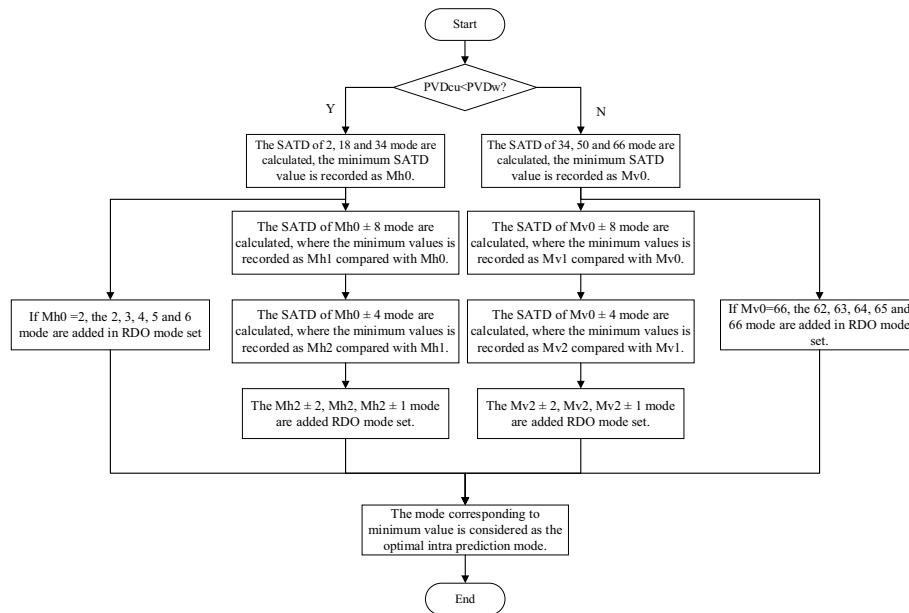
Fig. 7 The flowchart of the fast CU size decision algorithm

belong to the horizontal mode and 34–67 modes belong to the vertical mode. Specifically, the mean absolute deviation between pixels can more accurately represent the energy direction trend of a CU. Consequently, the pixel value deviation (PVD) technique is used for fast intra-prediction decision approach to obtain the texture direction of CU.  $PVD_D$  is denoted as

$$PVD_D(x) = \frac{1}{W \times H} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |P(n, i) - y(x)|, \quad x = 0, 1, 2, 3 \quad (29)$$

$$y(x) = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=1}^{H-1} P(x, i) \quad (30)$$

where  $y(x)$  represents the average luma value of the  $x$ -th pixel bar.  $W$  and  $H$  denote the width and height of CU, respectively.  $P(x, i)$  represents the  $i$ -th luma pixel value in the  $x$ -th pixel bar.  $PVD_w$  is originated from the average value of  $PVD_D$  of pixel bar that is defined as



**Fig. 8** The flowchart of the fast intra-prediction mode decision method

$$PVD_w = \frac{1}{N} \sum_{x=0}^{N-1} PVD(x) \tag{31}$$

where  $N$  refers to the number of the pixel bar. Specifically, the PVD value of CU compared with  $PVD_w$ , if it is less than  $PVD_w$ , the intra-prediction mode of CU is considered as the horizontal mode; otherwise, it belongs to the vertical mode.

If it belongs to horizontal mode, the SATD values of 2, 18, and 34 mode are first calculated, the smallest SATD value is recorded as  $M_{h0}$ . Then, the SATD values of  $M_{h0} \pm 8$  mode are calculated (if it does not exist, the calculation is not performed). In addition, the smallest SATD value in  $M_{h0} \pm 8$  mode is recorded as  $M_{h1}$  compared with  $M_{h0}$  mode. If  $M_{h0}$  is 2 mode, the 2, 3, 4, 5, and 6 mode are added in the RDO mode set. Further, the SATD values of  $M_{h1} \pm 4$  mode compare with  $M_{h1}$ , where the smallest SATD value is recorded as  $M_{h2}$ . Finally, the  $M_{h2} \pm 2$ ,  $M_{h2}$ ,  $M_{h2} \pm 1$  mode are added in the RDO mode set. Similarly, if it belongs to vertical mode, the SATD values of 34, 50, and 66 mode are calculated, the smallest SATD value is recorded as  $M_{v0}$ . Compared with  $M_{v0}$  mode, the smallest SATD value in  $M_{v0} \pm 8$  mode is recorded as  $M_{v1}$ . The smallest SATD value in  $M_{v1} \pm 4$  mode is recorded as  $M_{v2}$  compared with  $M_{v1}$ . If  $M_{v0}$  is 66 mode, 62, 63, 64, 65, and 66 modes are added in the RDO mode set. Finally,  $M_{v2} \pm 2$ ,  $M_{v2}$ ,  $M_{v2} \pm 1$  mode are added in the RDO mode set. The RDO value of each directional mode in the RDO mode set is computed, where the mode in correspondence with the minimum value is considered as the optimal mode.

#### 4 Experimental results and analysis

To access the coding property of the proposed overall algorithm, the test video sequences (A1, A2, B, C, D, E) are tested in “All intra” configuration. Bjontegaard Delta Bitrate (BDBR) [32] is utilized to gauge the overall encoding quality, while average time saving (ATS) measures the coding time saving achieved in H.266/VVC. In addition, if the BDBR increase is greater, the encoding quality is worse. In addition, the ATS is defined as

$$\text{ATS} = \frac{T_{\text{VTM7.0}} - T_{\text{Proposed}}}{T_{\text{VTM7.0}}} \times 100\% \quad (32)$$

where  $T_{\text{Proposed}}$  denotes the encoding runtime of the proposed algorithm,  $T_{\text{VTM7.0}}$  denotes the encoding time of the anchor method in VTM7.0.

The proposed overall algorithm consists of the fast CU size decision (FCSD) algorithm and the fast intra-prediction mode decision (FPMD) algorithm together, and Table 2 illustrates the ablation experimental results of the two individual methods. It can be found from Table 2 that the ATS is decreased by 39.86% and the increment of BDBR is ignorable in the FCSD algorithm. Experiment results show that the FCSD approach can early terminate CU splitting, thereby reducing unnecessary RDO calculations. It is found that the average coding runtime saving is 33.51% and the increment of BDBR is negligible in the FPMD algorithm. Thence, the FPMD algorithm can skip some needless intra-prediction modes, which can significantly reduce the encoding complexity.

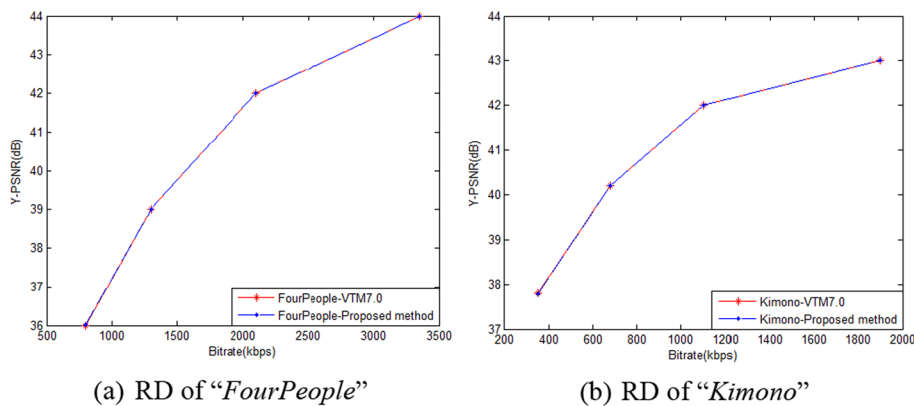
The results of the proposed overall algorithm are shown in Table 3. It is evident that the results demonstrate a 54.38% reduction in the encoding runtime of the proposed

**Table 2** Results of the proposed individual scheme

Sequences		FCSD		FPMD	
		BDBR (%)	ATS (%)	BDBR (%)	ATS (%)
Class A1 4 K	Tango2	0.57	41.43	0.49	31.79
	FoodMarket4	0.55	40.87	0.47	35.64
	Campfire	0.63	34.39	0.49	38.47
Class A2 4 K	Catrobot1	0.69	42.81	0.54	32.08
	DaylightRoad2	0.75	36.85	0.51	33.78
	ParkRunning3	0.68	42.37	0.52	36.82
Class B 1920 × 1080	Kimono	0.73	46.76	0.59	36.85
	ParkScene	0.68	40.62	0.42	35.68
	BQTerrace	0.78	39.65	0.45	30.99
Class C 832 × 480	PartyScene	0.72	38.77	0.53	38.71
	RaceHorsesC	0.79	40.21	0.51	31.79
	BasketballDrill	0.95	39.96	0.39	29.27
Class D 416 × 240	BlowingBubbles	0.68	40.14	0.41	31.45
	RaceHorses	0.66	43.66	0.51	31.02
	BQSquare	0.76	39.04	0.48	31.54
Class E 1280 × 720	Johnny	0.79	35.83	0.42	34.56
	FourPeople	0.77	37.46	0.53	30.49
	KristenAndSara	0.69	36.72	0.95	32.24
Average		0.72	39.86	0.51	33.51

**Table 3** Results of the proposed overall scheme

Sequences		Proposed	
		BDBR (%)	ATS (%)
Class A1 4 K	Tango2	0.99	56.85
	FoodMarket4	0.97	53.53
	Campfire	0.86	49.67
Class A2 4 K	Catrobot1	1.12	55.99
	DaylightRoad2	0.97	54.53
	ParkRunning3	1.09	53.57
Class B 1920 × 1080	Kimono	1.22	57.56
	ParkScene	0.95	59.22
	BQTerrace	1.29	52.41
Class C 832 × 480	PartyScene	0.95	53.66
	RaceHorsesC	0.91	53.75
	BasketballDrill	0.94	50.68
Class D 416 × 240	BlowingBubbles	0.91	54.88
	RaceHorses	1.01	49.45
	BQSquare	1.16	55.55
Class E 1280 × 720	Johnny	0.96	59.33
	FourPeople	0.89	58.72
	KristenAndSara	1.23	57.62
Average		1.02	54.83



**Fig. 9** RD performance comparison chart of the proposed overall algorithm

overall algorithm, and BDBR is increased by 1.02% compared with VTM 7.0. It is observed that the simulation results may fluctuate that is acceptable for videos with varying resolutions, as the impact of resolution on the results is minimal. Simulation results illustrate that the proposed overall method achieves a substantial reduction in encoding complexity while guaranteeing excellent encoding performance.

The RD performance comparison of two test video sequences "FourPeople" and "Kimono" is shown in Fig. 9. A comparison with VTM 7.0 reveals that the proposed overall algorithm exhibits nearly consistent RD performance.



**Table 4** Comparison of the proposed overall algorithm with the FQPD and ACSD algorithms

Sequences		FQPD[22] (VTM7.0)		ACSD[30] (VTM5.0)		Proposed (VTM7.0)	
		BDBR (%)	ATS (%)	BDBR (%)	ATS (%)	BDBR (%)	ATS (%)
Class B 1920 × 1080	Kimono	1.93	59.51	0.87	32.32	1.22	57.56
	ParkScene	1.26	51.84	0.83	35.41	0.95	59.22
	BQTerrace	1.08	45.30	0.95	34.50	1.29	52.41
Class C 832 × 480	PartyScene	0.26	38.62	0.55	31.10	0.95	53.66
	RaceHorsesC	0.88	49.05	0.37	23.63	0.91	53.75
	BasketballDrill	1.82	48.48	1.30	33.39	0.94	50.68
Class D 416 × 240	BlowingBubbles	0.47	40.35	0.95	33.90	0.91	54.88
	RaceHorses	0.54	41.69	0.71	31.79	1.01	49.45
	BQSquare	0.19	31.95	0.68	30.73	1.16	55.55
Class E 1280 × 720	Johnny	3.22	56.88	1.72	38.85	0.96	59.33
	FourPeople	2.70	57.57	1.35	38.01	0.89	58.72
	KristenAndSara	2.78	55.11	1.61	34.84	1.23	57.62
Average		1.43	48.03	0.99	33.21	1.04	55.24

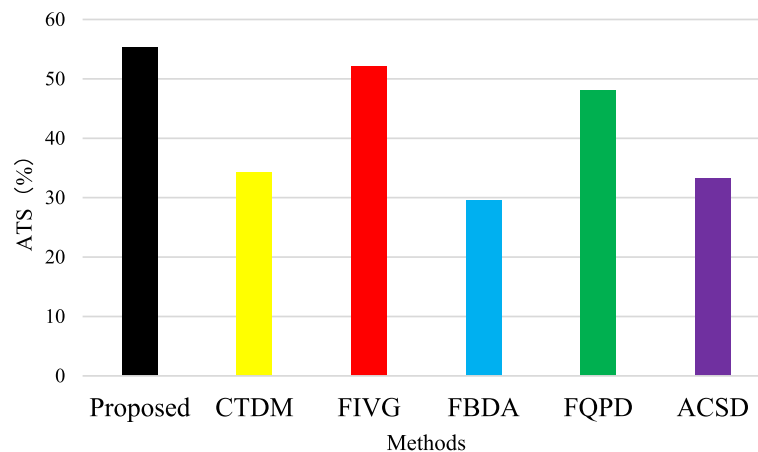
**Table 5** Comparison of the proposed overall algorithm with the CTDM, FIVG and FBDA algorithms

Sequences		CTDM [20] (VTM4.0)		FIVG [21] (VTM4.0)		FBDA [33] (VTM4.0)		Proposed (VTM7.0)	
		BDBR (%)	ATS (%)	BDBR (%)	ATS (%)	BDBR (%)	ATS (%)	BDBR (%)	ATS (%)
Class B 1920 × 1080	Kimono	1.05	34.33	1.72	66.59	1.98	41.82	1.22	57.56
	Park-Scene	1.11	34.56	1.28	56.28	1.38	31.60	0.95	59.22
	BQTerrace	1.00	31.07	1.16	49.44	1.19	29.47	1.29	52.41
Class C 832 × 480	Party-Scene	0.76	35.93	0.28	41.71	1.05	35.23	0.95	53.66
	Race-HorsesC	0.82	36.06	0.84	52.07	2.96	33.89	0.91	53.75
	BasketballDrill	1.67	33.94	1.91	53.05	1.36	28.73	0.94	50.68
Class D 416 × 240	Blowing-Bubbles	0.74	35.04	0.49	43.90	0.73	21.87	0.91	54.88
	Race-Horses	0.95	35.96	0.54	44.93	1.59	31.83	1.01	49.45
	BQSquare	0.61	33.98	0.17	32.34	−0.11	23.00	1.16	55.55
Class E 1280 × 720	Johnny	1.44	33.02	3.07	62.55	1.51	24.44	0.96	59.33
	FourPeople	1.38	35.05	2.55	62.18	1.37	26.65	0.89	58.72
	KristenAndSara	1.19	32.95	2.56	60.82	1.53	25.32	1.23	57.62
Average		1.06	34.32	1.38	52.16	1.38	29.49	1.04	55.24

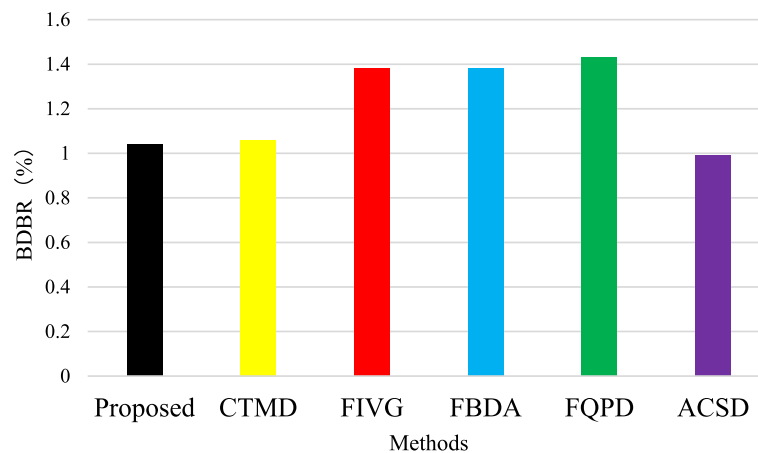
To assess the encoding performance of the proposed algorithm, we selected some classic fast and effective methods for comparison. The results of the comparison experiments with the five algorithms, CTDM [20], FIVG [21], FBDA [33], FQPD [22] and ACSD [30], are shown in Tables 4 and 5. The results of the advanced fast methods

including CTDM, FIVG, and FBDA method are compared with VTM4.0. In addition, ACSD and FQPD algorithms are compared with VTM 5.0 and VTM7.0 respectively. Tables 4 and 5 demonstrate that the proposed overall method achieves a coding time saving of 55.24% with a bit loss negligible. According to Tables 4 and 5, CTDM, FIVG, FBDA, FQPD, and ACSD methods all exhibit good coding characteristics, but in comparison, none of these methods save as much coding runtime as the proposed overall algorithm. The saved coding time increases by 20.92%, 3.08%, 25.75%, 7.21% and 22.03% compared to CTMD, FIVG, FBDA, FQPD and ACSD algorithms, respectively. That is, the proposed overall algorithm saves the most coding time. Furthermore, it can be observed from Tables 4 and 5 that the BDBR of the proposed algorithm increases by only 1.04%, while the BDBR of the CTDM, FIVG, FBDA, FQPD, and ACSD algorithms increases by 1.06%, 1.38%, 1.38%, 1.43%, and 0.99%, respectively. Thus, the BDBR of the proposed algorithm is much smaller than that of the CTDM, FIVG, FBDA, and FQPD algorithms, decreasing by 0.02%, 0.34%, 0.34%, and 0.39% compared to them, respectively. Although the average BDBR of the ACSD algorithm is smaller than that of the proposed algorithm in this paper, the difference is only 0.05, which is negligible. The FCSD and FPMD algorithms together constitute the overall algorithm proposed, where the FCSD algorithm can early terminate some needless CU partition modes and the FPMD algorithm can skip many needless prediction modes. Thus, the proposed overall scheme can decrease more encoding burden and ensure nearly consistent coding quality compared with the other algorithms in H.266/VVC.

The comparison of the algorithm proposed in this paper with other algorithms in terms of time saving and BDBR is shown more visually in Figs. 10 and 11. Figure 10 illustrates that the average runtime savings of the proposed overall algorithm are about 3.08–25.75% in comparison to the CTDM, FIVG, FBDA, FQPD, and ACSD algorithm. Furthermore, as depicted in Fig. 11, the proposed overall algorithm can reduce the BDBR by 0.02–0.39% compared to CTDM, FIVG, FBDA and FQPD algorithms. Simulation results show that the proposed overall algorithm saves more coding time and has superior coding performance compared to other fast and efficient algorithms, and is also applicable to video sequences with multiple resolutions.



**Fig. 10** Time saving



**Fig. 11** BDBR increase

## 5 Conclusions

To cope with the asymmetric partition problem brought by the QTMT splitting architecture, we propose a fast intra-CU size decision scheme and a fast intra-prediction mode decision algorithm. In the fast CU size decision scheme, the effective features are first chosen to decide CU splitting mode. Subsequently, the SVM models corresponding to CU size are trained on-line by using these features. Eventually, the SVM models are utilized for determining CU partition mode. In addition, the fast intra-prediction mode decision method is also devised, where the quantity of intra-prediction modes added to the RDO mode set decreases through an improved method. Simulation results show that the proposed overall algorithm reduces the coding runtime by 55.24% while maintaining stable coding quality (only 1.04% BDBR loss, which is negligible).

### Author contributions

Conceptualization, M.L. and Z.W.; methodology, M.L.; software, Z.W.; validation, M.L., Q.Z. and Z.W.; formal analysis, Z.W.; investigation, Z.W.; resources, Q.Z.; data curation, Z.W.; writing—original draft, Z.W.; writing—review and editing, M.L.; visualization, M.L.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, Q.Z. All the authors have read and agreed to the published version of the manuscript.

### Funding

This work was supported in part by the National Natural Science Foundation of China No.61771432, and 61302118, the Basic Research Projects of Education Department of Henan No. 21zx003, and the Key projects Natural Science Foundation of Henan 232300421150, the Scientific and Technological Project of Henan Province 232102211014, and the Postgraduate Education Reform and Quality Improvement Project of Henan Province YJS2021KC12, YJS2023JC08, and YJS2022AL034.

### Data availability

Not applicable.

### Declarations

#### Competing interests

The authors declare no conflict of interest.

Received: 17 October 2023 Accepted: 6 March 2024

Published online: 18 March 2024

## References

- B. Bross, K. Andersson, M. Bläser, V. Drugeon, S. Kim, J. Lainema, J. Li, S. Liu, J. Ohm, G.J. Sullivan, R. Yu, General video coding technology in responses to the joint call for proposals on video compression with capability beyond HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **30**(5), 1226–1240 (2020)
- J. Chen, M. Karczewicz, Y. Huang, K. Choi, J. Ohm, G.J. Sullivan, The joint exploration model (JEM) for video compression with capability beyond HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **30**(5), 1208–1225 (2020)
- B. Bross et al., Overview of the versatile video coding (VVC) standard and its applications. *IEEE Trans. Circuits Syst. Video Technol.* **31**(10), 3736–3764 (2021)
- Y.-W. Huang et al., Block partitioning structure in the VVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **31**(10), 3818–3833 (2021)
- Y.-W. Huang et al., A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques. *IEEE Trans. Circuits Syst. Video Technol.* **30**(5), 1311–1325 (2020)
- F. Bossen, K. Suhling, A. Wiecekowsky, S. Liu, VVC complexity and software implementation analysis. *IEEE Trans. Circuits Syst. Video Technol.* **31**(10), 3765–3778 (2021)
- C. Yao, C. Xu, M. Liu, RDNet: rate–distortion-based coding unit partition network for intra-prediction. *Electronics* **11**(6), 916 (2022)
- T. Wang, F. Li, X. Qiao, P.C. Cosman, Low-complexity error resilient HEVC video coding: a deep learning approach. *IEEE Trans. Image Process.* **30**, 1245–1260 (2021)
- H. Liu, R. Yang, S. Zhu, X. Wen, and B. Zeng, "Luminance-Guided Chrominance Image Enhancement for HEVC Intra Coding," in *Proc. 2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, (Austin, TX, USA, 2022), pp. 3180–3184
- J. Tariq et al., Nature inspired algorithm based fast intra mode decision in HEVC. *Multimed. Tools Appl.* **82**(19), 29789–29804 (2023)
- W. Imen, M. Amna, B. Fatma, S.F. Ezahra, N. Masmoudi, Fast HEVC intra-CU decision partition algorithm with modified LeNet-5 and AlexNet. *SIVIP* **16**(7), 1811–1819 (2022)
- V.V. Menon, H. Amirpour, C. Timmerer, and M. Ghanbari, "INCEPT: Intra CU Depth Prediction for HEVC," in *Proc. 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSp)*, (Tampere, Finland, 2021), pp. 1–6
- N. Elsayy, M.S. Sayed, F. Farag, Efficient coding unit classifier for HEVC screen content coding based on machine learning. *J. Real-Time Image Proc.* **19**(2), 375–390 (2022)
- M. Amna, W. Imen, and S. F. Ezahra, "Deep Learning For Intra Frame Coding," in *Proc. 2021 International Conference on Engineering and Emerging Technologies (ICEET)*, (Istanbul, Turkey, 2021), pp. 1–4
- J. Tariq, A. Armghan, A. Ijaz, I. Ashraf, Light weight model for intra mode selection in HEVC. *Multimed Tools Appl* **80**(14), 21449–21464 (2021)
- V. Galiano, H. Migallón, M. Martínez-Rach, O. López-Granado, M.P. Malumbres, On the use of deep learning and parallelism techniques to significantly reduce the HEVC intra-coding time. *J. Supercomput.* **79**(11), 11641–11659 (2023)
- Y.-H. Ho, C.-H. Kao, W.-H. Peng, and P.-C. Hsieh, "Neural Frank-Wolfe Policy Optimization for Region-of-Interest Intra-Frame Coding with HEVC/H.265," in *Proc. 2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, (Suzhou, China, 2022), pp. 1–5
- Q. Zhang, T. Cui, L. Huang, B. Jiang, J. Zhao, Low-complexity intra coding scheme based on Bayesian and L-BFGS for VVC. *Digital Signal Proc.* **127**, 103539 (2022)
- Q. He, W. Wu, L. Luo, C. Zhu, and H. Guo, "Random Forest Based Fast CU Partition for VVC Intra Coding," in *Proc. 2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, (Chengdu, China, 2021), pp. 1–4
- S. Park, J. Kang, Context-based ternary tree decision method in versatile video coding for fast intra coding. *IEEE Access* **7**, 172597–172605 (2019)
- J. Chen, H. Sun, J. Katto, X. Zeng and Y. Fan, "Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient," in *Proc. 2019 IEEE Visual Communications and Image Processing (VCIP)*, (Sydney, Australia, 2019), pp. 1–4
- Y. Fan, J. Chen, H. Sun, J. Katto, M. Jing, A fast QTMT partition decision strategy for VVC intra prediction. *IEEE Access* **8**, 107900–107911 (2020)
- M. Saldanha, G. Sanchez, C. Marcon, L. Agostini, Configurable fast block partitioning for VVC intra coding using light gradient boosting machine. *IEEE Trans. Circuits Syst. Video Technol.* **32**(6), 3947–3960 (2022)
- H. Liu, S. Zhu, R. Xiong, G. Liu, and B. Zeng, "Cross-Block Difference Guided Fast CU Partition for VVC Intra Coding," in *Proc. 2021 International Conference on Visual Communications and Image Processing (VCIP)*, (Munich, Germany, 2021), pp. 1–5
- T. Li, M. Xu, R. Tang, Y. Chen, Q. Xing, DeepQTMT: a deep learning approach for fast QTMT-based CU partition of intra-mode VVC. *IEEE Trans. Image Process.* **30**, 5377–5390 (2021)
- Y. Li, G. Yang, Y. Song, H. Zhang, X. Ding, D. Zhang, Early intra CU size decision for versatile video coding based on a tunable decision model. *IEEE Trans. Broadcast.* **67**(3), 710–720 (2021)
- M.-J. Chen et al., Efficient partition decision based on visual perception and machine learning for H.266/versatile video coding. *IEEE Access* **10**, 42141–42150 (2022)
- S. Wu, J. Shi, Z. Chen, HG-FCN: hierarchical grid fully convolutional network for fast VVC intra coding. *IEEE Trans. Circuits Syst. Video Technol.* **32**(8), 5638–5649 (2022)
- F. Wang, Z. Wang, Q. Zhang, FSVM- and DAG-SVM-based fast cu-partitioning algorithm for VVC intra-coding. *Symmetry* **15**(5), 1078 (2023)
- G. Tang, M. Jing, X. Zeng and Y. Fan, "Adaptive CU split decision with pooling-variable CNN for VVC intra encoding," in *Proc. 2019 IEEE Visual Communications and Image Processing (VCIP)*, (Sydney, Australia, 2019), pp. 1–4
- S. De-Luxan-Hernandez, V. George, J. Ma, T. Nguyen, H. Schwarz, D. Marpe, and T. Wiegand, "An intra subpartition coding mode for VVC," in *Proc. 2019 IEEE International Conference on Image Processing (ICIP)*, (Taipei, Taiwan, 2019), pp. 1203–1207

32. G. Bjontegaard, Calculation of Average PSNR Differences Between RD Curves, document ITU-T SG16 Q6, VCEG-M33, Austin, TX, USA, (2001)
33. N. Tang, J. Cao, F. Liang, J. Wang, H. Liu, X. Wang and X. Du, "Fast CTU Partition Decision Algorithm for VVC Intra and Inter Coding," in *Proc. 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, (Bangkok, Thailand, 2019), pp. 361–364

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.