IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 October 2023, accepted 22 October 2023, date of publication 25 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327519

RESEARCH ARTICLE

Fast Adaptive CU Partition Decision Algorithm for VVC Intra Coding

LINA SI, WENDI ZHU, AND QIUWEN ZHANG
College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Corresponding author: Qiuwen Zhang (zhangqwen@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771432 and Grant 61302118; in part by the Basic Research Projects of Education Department of Henan under Grant 21zx003 and Grant 20A880004; in part by the Key Projects of Natural Science Foundation of Henan under Grant 232300421150; in part by the Scientific and Technological Project of Henan Province under Grant 232102211014; and in part by the Postgraduate Education Reform and Quality Improvement Project of Henan Province under Grant YJS2021KC12, Grant YJS2023JC08, and Grant YJS2022AL034.

ABSTRACT The latest video standard - Versatile Video Coding Standard (VVC/H.266) has been standardized and officially entered into force. Compared with the High Efficiency Video Coding (HEVC/H.265), owing to the introduction of the Quad-tree with Nested Multi-type Tree (QTMT) division mode, the encoder can choose a more detailed division type when dividing the Coding unit (CU), thereby improving the coding performance. However, when the CU selects the division type, it needs to traverse all possible division types and compute the Rate-distortion (RD) cost, which greatly enhances the coding complexity. Therefore, this paper designs a Joint Random Forest Classifier (JRFC) to make decisions on CU partition types, and proposes fast adaptive CU partition decision algorithm for VVC intra coding combined with our previous work. The algorithm has the capability to make partition decisions for CUs of diverse sizes (smaller than 32×32) and completely bypass the intricate process of Rate-distortion Optimization (RDO), resulting in a significant reduction in encoding time. The experimental results demonstrate that, in comparison to the basic approach utilized in VTM10.0, the algorithm proposed in this paper reduces the encoding time by 57.27% on average, with only a marginal increase of 1.53% in Bjøntegaard delta bit rate (BDBR).

INDEX TERMS VVC, QTMT, JRFC, feature extraction, fast CU partition algorithm.

I. INTRODUCTION

In this era of exponential explosion of information, video, as a carrier of information, has become a part of our daily life through different channels. So as to decrease the capacity of data occupied by video during compression and transmission, and facilitate the storage and transmission of video, video coding technology has gradually become a more popular research direction [1].

H.264/AVC [2] stands as a compressed digital video codec standard advanced by the Joint Video Team (JVT). Embracing cutting edge video coding technology, it endeavors to compress video data to its utmost capacity, all the while assuring unfaltering video quality. However, the H.264/AVC standard exhibits certain shortcomings when it comes to encoding high-definition video. Therefore, the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Professional Group (MPEG) established a Video Coding Joint Collaboration Team JCT -VC, jointly develop a more efficient video coding standard - H.265/HEVC [3]. This new standard can more effectively compress high-definition video data, thereby providing higher-quality video content, and providing better support for high-definition video applications. The H.264/AVC and H.265/HEVC standards have brought significant technological advances in video compression, making efficient encoding of high-resolution video a reality. However, under current video applications and industry demands, the performance of these two standards still has not reached the expected coding efficiency, so continuous improvement is needed, which also promotes the establishment of new video coding standards by international organizations. In July 2020, the Joint Video Experts Team

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

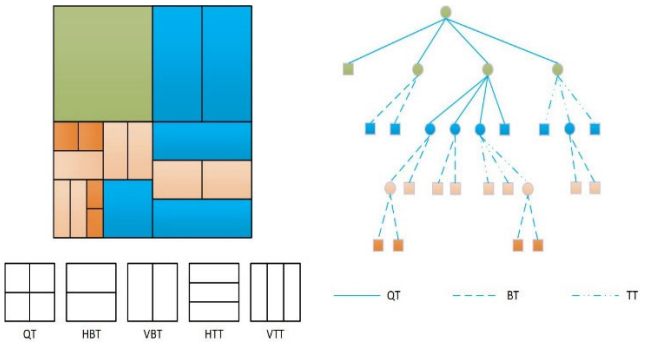
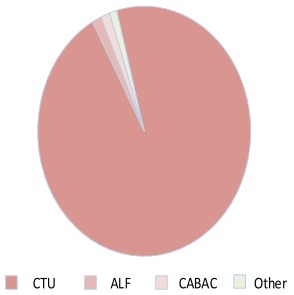


FIGURE 1. QTMT division process in VVC.



Key technology	Encoding time	Proportion
CTU	5689.1 s	96.54 %
ALF	92.4 s	1.57 %
CABAC	58.9 s	1.00 %
Other	52.9 s	0.89 %
Total encoding time	5893.3 s	100%

FIGURE 2. Statistics chart of encoding complexity. (Take the BQMall sequence as an example.)

(JVET) officially released the Versatile Video Coding (VVC) [4] standard, marking the official launch of a new generation of video coding standards.

Compared with HEVC, the newly formulated VVC has more efficient coding performance and broad market prospects. In VVC, many novel technologies are adopted: Quad-tree with Nested Multi-type Tree (QTMT) block division mode, support for 65 intra prediction modes and so on [5]. The primary contribution to the enhancement of coding performance lies in the adoption of a more versatile and pliable Coding Unit (CU) partition structure known as QTMT partition. That is, on the foundation of the original division type of HEVC - Quad-tree (QT) division, Binary-tree (BT) division and Ternary-tree (TT) division are introduced, which are subdivided into four types of divisions, namely: Horizontal Trinomial-tree (HTT), Vertical Trinomial-tree (VTT), Horizontal Binary-tree (HBT), and Vertical Binary-tree (VBT), the specific information is reflected in Figure 1. [6]. Same as HEVC, Rate-distortion Optimization (RDO) is also used in VVC as the basis for the selection of Coding Tree Unit (CTU) division types. The prescribed procedure unfolds as follows: the encoder meticulously navigates through an array of potential division types, sequentially computing their Rate-distortion (RD) costs, and ultimately discerns the division type endowed with the most meager RD cost, thus anointing it as the optimal division method for the Coding Unit (CU) [7]. Whilst RDO possesses the capability to elect the most exemplary division mode for Coding Units (CUs) and thereby enhance coding performance, the proliferation of division types within the Versatile Video Coding (VVC) standard necessitates the computation of RD costs for every conceivable division of a CU. This arduous task bestows upon us an inconceivable magnitude of calculations. The formula employed to ascertain the RD cost and Sum of Absolute Difference (SAD) is as follows:

$$SAD\,(x, y) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left| f_i\,(m, n) - f_{i-1}\,(m + x, n + y) \right|, \quad (1)$$

where x represents the horizontal component, y represents the vertical component, and the RD cost criterion is calculated as follows:

$$J = SAD\,(x, y) + \lambda \cdot R, \quad (2)$$

Among them, λ represents the lagrangian factor, and R represents the number of bits required for encoding.

We encode the first 50 frames of the BQMall sequence under the All Intra (AI) configuration of the VTM10.0 encoder, and count the encoding time used, the specific information is reflected in Figure 2. It shows that among the key technologies of VVC, the CTU division process takes the most time, with a proportion as high as 96.54%. Therefore, we can conclude that the QTMT technology provides a flexible division method for CTU division not only to improve the coding performance, but occupies the most coding time of VVC.

Following the aforementioned theoretical analysis and research experiments, it becomes apparent that the advancement of Coding Unit (CU) partition technology not only amplifies the coding quality, but also bestows upon us a formidable computational complexity, thereby setting forth elevated demands on the implementation of coding and decoding processes. Henceforth, it is imperative to discover an expeditious algorithm to supplant the laborious Rate-Distortion Optimization (RDO) process and accelerate the partitioning of Coding Units (CUs), thereby diminishing the intricacies entailed in coding and decoding procedures. In the traditional encoding method, the partition mode of each CU is first initialized in the block partition mode list, and all the partition modes that can be tried are written into the list, and then all the partition modes in the list will be tried in the actual partition process. It can be seen that this process will cause a lot of computational complexity. Therefore, in fast algorithm design, partition methods with low probability can be predetermined and removed from the initialization list. In our antecedent endeavor [8], we presented a method to diminish the encoding time in Versatile Video Coding (VVC) through a fusion of statistical analysis and the SAE-CNN. This method adeptly formulates partition decisions for Coding Units (CUs) of varying dimensions. However, the previous work can only decide whether to divide the CU, and the specific division type decision still needs to use the RDO process of the traditional algorithm, which needs to be further improved. Henceforth, the present study endeavors to devise a Joint Random Forest Classifier (JRFC) to undertake the intricate task of decision-making pertaining to the partition type of Coding Units (CUs). Through a seamless integration with our preceding research [8], we culminate in proposing

a swift and adaptable algorithm for CU partition decision in the realm of VVC intra coding. The algorithm possesses the capability to handle Coding Units (CUs) of diverse sizes resulting from the QTMT structure. It aptly formulates partition decisions for CUs with dimensions below 32 × 32, thereby bypassing the exhaustive Rate-Distortion Optimization (RDO) process, thus engendering a substantial reduction in encoding time. This approach achieves a commendable balance between the mitigation of computational complexity and the compromise in encoding performance.

II. PREPARATION WORKS

A. FAST ALGORITHMS FOR PREVIOUS STANDARDS

Prior to the formulation of VVC, research predominantly focused on the preceding standard, HEVC. Numerous expeditious algorithms leveraging machine learning techniques were proposed for HEVC, aiming to expedite CU partitioning and mitigate computational complexity. Zhang et al. [9] proposed a Convolutional Neural Network (CNN)-based algorithm for determining the partitioning of coding units (CUs). Treated CNN as a classifier, the algorithm inputs a CU of size 64 64, and finally obtains a sequence consisting of 21 flag bit information. This sequence represents the overall division trend of CU, and three division schemes are proposed to make decisions on CU division. Xu et al. [10] first proposed a pre-partition depth prediction based on Quantization Parameter (QP) and texture complexity, and designed a NN classifier to transform the CU division decision into a binary classification issue to speed up the inter coding process. In [11], a Support Vector Machine (SVM) based HEVC intra prediction complexity reduction algorithm is proposed. By using an SVM classifier, the variance and low frequency AC coefficients components are selected as feature inputs to quickly determine the CU size. In [12], the matter of determining the depth of coding units (CUs) is approached as a hierarchical decision problem with three layers. An algorithm for CU depth decision is proposed, wherein the authors devise a classifier with three distinct output outcomes, aiming to mitigate the risk of misprediction and enhance the accuracy of predictions. Zhang et al. [13] proposed an enhanced RMD process that groups the 35 predicted modes based on their phase angles, thereby reducing the number of candidate modes.

The above-mentioned fast algorithm of HEVC based on machine learning shows good performance of reducing complexity in the actual coding process. However, since VVC introduces more division types in block division, the algorithm proposed for HEVC cannot be used well to accelerate the CU division process of VVC.

B. FAST ALGORITHMS FOR VVC

With the completion of the VVC standardization work, although VVC showed better coding performance, it also led to a substantial increase in coding time. Therefore, researchers began to explore the VVC fast CU partition algorithm that reduces coding complexity. Methods for streamlining the coding tree unit (CTU) partitioning in VVC can typically be classified into two overarching categories: heuristic approaches and data-driven methodologies.

For data-driven approaches, in [14], a compelling algorithm founded up-on Convolutional Neural Networks (CNN) is brought forth, purposed to mitigate the complexity entailed in VVC intra-coding. The algorithm sends 64 64 CUs as input to CNN, and outputs 480 prediction vectors representing the probability of boundary division. Guided by the output vectors, the encoder possesses the acumen to circumvent the intricate process of Rate-Distortion Optimization (RDO) and seamlessly proceed with the division of Coding Units (CUs). Li et al. [15] ingeniously devised a multi-phased exit CNN model, harnessing the principle of early termination to expedite the process of Coding Unit (CU) partitioning. Additionally, they put forth a method of multi-threshold decision, ingeniously tailored to enhance the predictive capabilities of the model. In [16], an intricacy mitigation algorithm, hinged upon Convolutional Neural Networks (CNN) and Random Forest Classifier (RFC), is introduced. This algorithm artfully amalgamates the predictive powers of CNN and RFC to discern the depth and division type of a 32×32 Coding Unit (CU). This discernment empowers the algorithm to circumvent the arduous process of RDO, effectively reducing encoding time. Jing et al. [17] combined traditional methods and deep learning methods, and proposed a technique to simplify VVC intra prediction through gradient analysis and multi-feature fusion CNN.

For heuristic approaches, Chen et al. [18] used distributions representing human visual perceptual information as input features for machine learning. A fast MTT decision based on machine learning random forest model is proposed to quickly select partitions for intra coding. Zhang et al. [19] proposed an expeditious algorithm for Coding Unit (CU) partition and intra mode decision, wherein they intricately crafted a Random Forest Classifier (RFC) model to discern the optimal CU division type. Moreover, they judiciously optimized the intra prediction mode by leveraging the distinctive attributes of texture regions. Fu et al. [20] suggested a CU vertical division skipping algorithm using Bayesian decision, and took the sub- CU partition type and intra prediction mode as features to reduce the computational complexity. In [21], an RFC-based fast CU partitioning algorithm is proposed, which firstly divides CUs into three categories: ''simple'', ''complex'', and ''fuzzy'', according to the contrast of texture complexity. Then two classifiers are designed for different CU categories to speed up the division process. In [22] and [23], RFC-based fast QTBT partitioning decisions are proposed, and a classifier is designed to decide whether the encoder should do QT or BT partitioning for CUs. At the same time, the authors introduce RD error to evaluate the impact of misclassification on coding efficiency. While the aforementioned expedited algorithms exhibit a measure of efficacy in mitigating coding complexity, their impact on performance remains somewhat inconspicuous, failing to strike an effective equilibrium between complexity reduction

and encoding performance cost. Zhao et al. [24] proposed a method to speed up the CU partition decision by training two SVM models. More reliable correlation features are selected based on the maximum ratio of standard deviation (SD) and edge point ratio (EPR) in sub-CUs.

C. MOTIVATION

In the past work [8] we proposed a VVC intra prediction technology, we use pre-decision and SAE-CNN to predict whether the coding unit is divided, which can reduce part of the RDO, so as to achieve the purpose of reducing the encoding time. Since the original block can reflect the characteristics of the block to a certain extent, many current algorithms use the original block as the input of CNN to speed up CU division. However, compared with the original block, the residual block (the CU that has been divided at least once) can better reflect the texture details of the image and the CU division trend. Therefore, if the residual coding unit is used as the input of CNN to make the division decision, more accurate prediction effect will be obtained and the prediction accuracy will be improved. It is precisely this motivation that we proposed in our past work.

But the scheme proposed in [8] can decide whether to partition CUs of different sizes, but does not fully consider its partition type decision. The core idea of the traditional heuristic fast algorithm is to manually select some features and manually set the corresponding discriminant criteria to predict the partition mode of the CU. However, according to the analysis, it can be seen that the block division structure of VVC is relatively complex, and there are many factors affecting CU division, and it is difficult to bring accurate prediction results by manually setting the discrimination criteria. Machine learning algorithm is an algorithm that automatically learns useful information from data. Due to its strong learning ability and generalization ability, it is widely used in various fields and promotes the development of various fields. Therefore, in order to improve the previous work and further judge the division category of CU, this paper proposes a fast CU division decision algorithm based on random forest, and uses random forest classifier to predict the division mode of CU.

III. PROPOSED ALGORITHM

A. JOINT RANDOM FOREST CLASSIFIER

The core idea of the traditional heuristic fast algorithm is to manually select some features and manually set the corresponding discrimination criteria to predict the CU division mode. However, according to the analysis, it can be seen that the block division structure of VVC is relatively complex and there are many factors that affect CU division. Manually setting the discrimination criteria is difficult to achieve accurate prediction results. Machine learning algorithm is an algorithm that automatically learns useful information from data. Manually extracted features are helpful for further research. Its strong learning ability and generalization ability

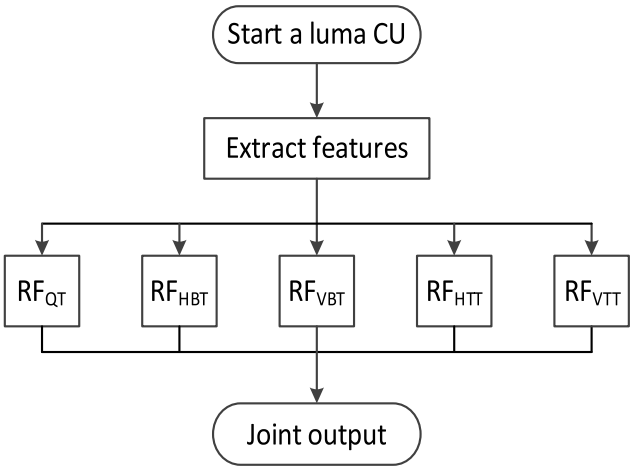


FIGURE 3. Diagram of joint random forest classifier framework.

are widely used in various fields. As can be seen from Table 3, the classification accuracy of the model proposed in this article is above 90%, and the classification results can accurately reflect the final division of coding units. (The accuracy of the RFC models in [16] and [17] is also above 90%) Therefore, the RFC acceleration algorithm proposed in this article has a certain degree of reliability.

Due to the superior performance of RFC in classification, this paper models the CU division type selection issue as a classification problem, and designs a JRFC to make decisions on the CU partition type. The biggest highlight of JRFC is that it consists of five independent binary RFCs: $\boldsymbol{RF_{QT}}, \boldsymbol{RF_{HBT}}, \boldsymbol{RF_{VBT}}, \boldsymbol{RF_{HTT}}, \boldsymbol{RF_{VTT}}$ and each classifier runs independently. Among them, $\boldsymbol{RF_i}$ represents the independent binary classifier, and $\boldsymbol{i} \in (\boldsymbol{QT}, \boldsymbol{HBT}, \boldsymbol{VBT}, \boldsymbol{HTT}, \boldsymbol{VTT})$ denotes the type of division that the classifier can make decisions. The algorithm flow is: input a luminance CU (the CU can be any size below 32×32), and perform appropriate feature extraction on the CU. The proposed algorithm does not process CUs with a size greater than 32×32, and encodes them according to the traditional process. Then, the extracted feature sequences are sent to five independent binary RFCs, and finally a sequence consisting of five data is output. According to the output sequence, the final classification result of JRFC can be obtained. The encoder divides the CU according to the classification result. Thus, the decision process of the division type of the CU is completed. The JRFC framework proposed in this paper is shown in Figure 3.

B. FEATURE SELECTION

For a classifier, whether it is SVM or RFC, the most important step is the selection of features. Selecting appropriate features as the nodes of the decision tree can better complete the subsequent RFC training process, improve the prediction accuracy, and obtain better performance. In this work, for the proposed JRFC and its implemented functions, we select features from the following aspects, including QP, texture

TABLE 1. Table of partition types for CUs of different sizes.

QP	Partition Type (%)					
	QT	HBT	VBT	HTT	VTT	NOT
22	26.16	20.25	20.10	7.93	6.94	18.62
27	17.30	22.38	16.86	7.48	4.78	31.20
32	12.87	21.66	15.41	7.76	5.14	37.19
37	9.31	22.07	16.06	7.21	4.19	41.16

complexity, gradient information, sub-CUs complexity difference and block information.

QP: We select VVC standard video sequences with different texture complexity and different resolutions (*Drums*, *Tango*, *RollerCoaster*, *BasketballDrive*), and encode them under the AI configuration of the latest encoder version VTM10.0, and obtain the selection of CU division types of various sizes under different QP value encoding environments, and the percentage of each division type selected by CUs under different QP value coding is counted. For details, refer to Table 1.

From Table 1, we can see that with the increase of QP, the proportion of CU's "QT split" gradually decreases (from 26.16% to 9.31%), while the proportion of "Not split" gradually increases (from 18.62% to 41.16%), and the rest of the division types are roughly unchanged. This shows that the difference of the QP value during encoding is closely related to the CU division, and affects the selection of the CU division type to a certain degree. Therefore, we choose the QP value as one of the features.

Texture Complexity: Texture complexity reflects the complexity of information contained in a CU, and is usually used as an important feature when using a classifier to make CU partition decisions. In general, the higher the texture complexity is, the more detailed the CU will be divided, such as QT division. The lower the texture complexity is, the CU will perform rough division, such as TT, BT division, or even no division. Therefore, taking texture complexity as an input feature is of great significance to improving the prediction accuracy of RFC and reducing encoding time. Below we discuss several common eigenvalues for measuring texture complexity, and explore the best features suitable for the JRFC proposed in this paper as feature inputs.

In previous studies, people usually use the variance (Var) of the pixel values contained in the CU to measure the texture complexity of a CU. As shown in Eq. (3):

$$Var = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} [P(i,j) - (\frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} P(i,j))]^2 \tag{3}$$

FIGURE 4. Texture complexity comparison map of CUs with the same variance.

where P (i, j) represents the pixel value on the (i, j) coordinate within the CU; W indicates the width of the CU, and similarly H indicates the height.

Although the variance can well embody the texture information of the global CU, it cannot clearly embody the partial features of the CU. As shown in Figure 4, many CUs with the same variance have great differences in their local texture complexity. Therefore, variance cannot be used as a measure of complexity in this paper. People further proposed another feature to measure the complexity of CU texture – the mean of absolute difference between pixels ($P_{MADP}(i,j)$). The calculation formula is shown in Eq. (4):

$$P_{MADP}(i,j) = \frac{1}{8} \begin{pmatrix} |P(i,j) - P(i-1,j-1)| + |P(i,j) - P(i,j-1)| \\ +|P(i,j) - P(i+1,j-1)| + |P(i,j) - P(i+1,j)| \\ +|P(i,j) - P(i+1,j+1)| + |P(i,j) - P(i,j+1)| \\ +|P(i,j) - P(i-1,j+1)| + |P(i,j) - P(i-1,j)| \end{pmatrix} \tag{4}$$

where P (i, j) indicates the pixel value on the (i, j) coordinate within the CU.

On top of the existing features used to measure texture complexity, we conduct a more in-depth exploration, aiming to find features that can more exactly reflect the texture complexity of CU. First, the variance can roughly reflect the complexity information of a CU. We multiply the CU variance and the average of the absolute differences between pixels to obtain the relative texture complexity. The formula is shown in Eq. (5):

$$RPC(i,j) = P_{MADP}(i,j) \times Var \tag{5}$$

where RPC denotes the relative texture complexity; (i, j) represents the coordinate information of each pixel contained in the CU.

Next, on the basis of the original variance formula, we replace the pixel value in the formula with RFC to obtain the final feature used in this article to measure the CU texture complexity (TC). The formula is shown in Eq. (6):

$$TC = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \times \left[RPC\,(i,j) - \left(\frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} RPC(i,j \right) \right] \tag{6}$$

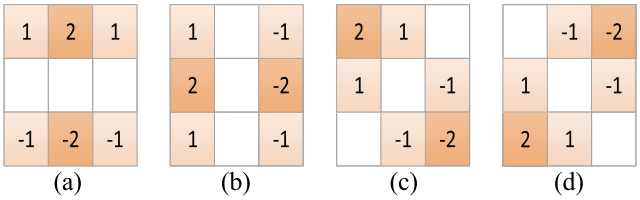


FIGURE 5. Soble operator convolution kernels in four directions. (a) Horizontal direction; (b) Vertical direction; (c) 45° direction; (d) 135° direction.

where H indicates the height of the CU, W indicates the width; (i, j) indicates the position of each pixel of the CU.

Compared with Var, TC replaces the pixel value in the original formula with on the basis of Var, so that TC can more exactly reflect the texture complexity of CU. In [20], a similar calculation method was also applied to the acceleration algorithm of HEVC and proved to be effective. To sum up, we finally choose TC as the feature input to measure the CU texture complexity.

Gradient Information: The gradient information reflects the change rate of image pixels, and is also closely related to the choice of CU partition type. Therefore, in this paper, we select 8 gradient values as features, namely: horizontal gradient ($Grad_x$), vertical gradient ($Grad_y$), 45-degree directional gradient ($Grad_{45°}$), 135-degree directional gradient ($Grad_{135°}$), average gradient ($Grad_{avg}$), maximum gradient ($Grad_{max}$) and gradient ratio ($Ratio_{x,y}$, $Ratio_{45°,135°}$,). We use the Sobel operator to convolve the target CU to get the gradient information we want. The calculation formula is shown in Eq. (7). It should be noted that the gradient information here is calculated on the global CU, not the sub-CUs.

$$Grad_o = \sum_{i=1}^{W}\sum_{j=1}^{H} P * S_o \,, o \in \{x, y, 45^o, 135^o\} \quad (7)$$

$$Grad_{avg} = \frac{1}{4}\sum Grad_o, \; o \in \{x, y, 45^o, 135^o\} \quad (8)$$

$$Grad_{\max} = \arg\max(Grad_o), \; o \in \{x, y, 45^o, 135^o\} \quad (9)$$

$$Ratio_{x,y} = \frac{Grad_x}{Grad_y} \quad (10)$$

$$Ratio_{45^o,135^o} = \frac{Grad_{45^o}}{Grad_{135^o}} \quad (11)$$

where P denotes the pixel array of the global CU; $S_O, o \in \{x, y, 45°, 135°\}$, denotes the convolution kernel of the Sobel operator in four directions, as shown in Figure 5; W indicates the width of the CU, and similarly H indicates the height.

Sub-CUs Complexity Difference: In order to achieve a better classification effect, we consider the complexity difference of sub-CUs as feature input as well. Here we use the average of the TC differences of the sub-CUs to describe the complexity difference of sub-CUs for each possible partition type. STC_{QT}, STC_{HBT}, STC_{VBT}, STC_{HTT}, STC_{VTT}, are calculated, the formula is as follows:

$$\begin{cases} STC_{QT} = \frac{1}{4}\sum_{i=1}^{4}(TC_i - \overline{TC_{QT}})^2 \\ STC_{HBT} = \frac{1}{2}\sum_{i=1}^{2}(TC_i - \overline{TC_{HBT}})^2 \\ STC_{VBT} = \frac{1}{2}\sum_{i=1}^{2}(TC_i - \overline{TC_{VBT}})^2 \\ STC_{HTT} = \frac{1}{3}\sum_{i=1}^{3}(TC_i - \overline{TC_{HTT}})^2 \\ STC_{VTT} = \frac{1}{3}\sum_{i=1}^{3}(TC_i - \overline{TC_{VTT}})^2 \end{cases} \quad (12)$$

where STC_i, i ∈ (QT, HBT, VBT, HTT, VTT), represents the complexity difference of sub-CUs under different partition types; TC_i indicates the texture complexity of the i -th sub-CUs after being divided, which is calculated by Eq. (6); TC denotes the average texture complexity of all sub-CUs after being divided.

Block Information: Because there are some prescribed division restrictions in the QTMT scheme, for example: HBT division in the second sub-CU of HTT division is disabled to prevent overlapping with two binary tree divisions; The VBT division in the second sub-CU of the VTT division is disabled to prevent overlapping with the two binary tree divisions; When the first sub-CU of the VBT partition is further divided into HTT, the HTT partition in the second sub-CU is disabled to prevent overlapping with binary tree and ternary tree partitions. Therefore, in order to prevent the occurrence of the above situation from causing coding errors, we also take the CU size as the feature input of the classifier.

C. MODEL TRAINING

The data samples used to train JRFC are basically same as those used to train SAE-CNN in our previous work [8]. The difference from constructing the SAE-CNN dataset is that we no longer categorize the training set according to the size of the CU. Instead, only all CUs that will be divided are selected, the 17 features (as shown in Table 2). described above are extracted from each CU as input, and the division type of the CU is marked, classified according to the QP value (in order to reduce the training complexity, we select four common QP values, namely QP=22, 27, 32, 37), and finally 4 data sets are formed.

In terms of training scheme, we do not directly train the proposed JRFC, but use grid search to train each independent binary RFC, and then compose the trained RFC into JRFC to complete the training process. Specifically, first select the data to set with a QP value of 22 to train RF_{QT}, and set the training parameter $n\_estimators$ to 100 to achieve the best prediction effect. RF_{QT} is then trained on datasets with QP values of 27, 32, and 37, respectively, with the parameters kept the same. By analogy, the training of the remaining

TABLE 2. The 17 features.

Feature Extraction	
QP	/
Texture Complexity	TC
Gradient Information	$Grad_x$、$Grad_y$、$Grad_{45°}$、$Grad_{135°}$、$Grad_{avg}$、$Grad_{max}$、$Ratio_{x,y}$、$Ratio_{45°,\ 135°}$
Sub-CUs Complexity Difference	STC_{HBT}、STC_{VBT}、STC_{HTT}、STC_{VTT}、STC_{QT}
Block Information	W、H

TABLE 3. Joint RFC performance test table.

Sequence	Prediction Accuracy (%)				Average (%)
	QP=22	QP=27	QP=32	QP=37	
Catrobot	89.76	92.32	90.31	95.87	92.07
Kimono	88.24	91.86	89.97	92.65	90.68
FourPeople	90.14	92.43	91.56	90.36	91.12

4 binary RFCs is completed. That is to say, we need to use 4 QP datasets to train 5 binary RFCs respectively, and a total of 20 binary classifiers need to be trained to form JRFC. Furthermore, we set a threshold (0.9) to binarize the output sequence of JRFC to avoid misclassification and improve the accuracy of CU partition type decision.

After completing the training process of the joint RFC, we use 3 VVC standard test sequences except the training samples to test the prediction accuracy of the JRFC model in terms of CU partition type decision under different QPs values. Please refer to Table 3 for the test results. According to the data in the table, we can clearly see that the average prediction accuracy of JRFC is above 90%, for different sequences. And in the same sequence, with different QPs values, the prediction accuracy remains roughly the same, with the highest being 95.87% and the lowest being 88.24% (within the allowable range). Therefore, we can conclude that the JRFC proposed in this paper has excellent classification performance and can be applied to make decisions on the CU partition type to speed up the CU partition process.

D. OVERALL ALGORITHM

Step 1) Input a luma CU, calculate the entropy value of the CU residual block and the QP value of the current encoder.

Step 2) Find the corresponding marked result in the pre-decision dictionary for the value obtained in step 1. If it is marked as ''Split'', skip step 3 and proceed directly to step 4 to complete the division decision-making process; If it is marked as ''Non_split'', the division decision procedure of the present CU can be ended in advance; If it is marked as ''Uncertain'', go to step 3.

Step 3) The residual block of the CU marked as ''Uncertain'' in step 2 is used as input, and sent to SAE-CNN to output the probability of dividing and not dividing the CU, and determine whether to partition the CU according to the probability. If the output result is ''Split'', go to step 4, select the appropriate division type, and complete the division decision-making process; If the output result is ''Non_split'', step 4 is skipped, and the division decision procedure of the present CU is ended in advance.

Step 4) The feature of the CU whose result is ''Split'' in step 3 is extracted as input, and sent to JRFC for division type decision, and the optimal division type of the CU can be obtained according to the output result, thereby completing the CU division decision process.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL ENVIRONMENT

The comprehensive experiment is conducted within the AI (All Intra) configured environment of the most recent encoder version, VTM10.0, of the VVC standard. And the construction and training process of JRFC as well as each encoding run on Intel Core i5-8500 CPU@3.00GHz processor on x64-based Window10 operating system. We have carefully chosen 19 VVC standard test sequences encompassing diverse resolutions and a wide range of video information. These sequences are categorized into six distinct groups: A1 (4K), A2 (4K), B (1920 × 1080), C (832 × 480), D (416 × 240), and E (1280 × 720). They constitute our comprehensive test set, which we employ to evaluate the efficacy of the proposed expeditious approach outlined in this study. We conducted an extensive evaluation of the test set within the coding environment, employing varying QP values (22, 27, 32, 37). It is worth noting that the construction and training of SAE-CNN and JRFC are considered offline processes, and thus, the time spent on these procedures is not factored into the encoding time calculations. Furthermore, in order to ensure the utmost efficacy and impartiality of the method, the sequences employed in the test set are entirely distinct from those utilized in the training set.

Here, we choose two criteria to measure the performance of the proposed algorithm, Bjøntegaard Delta Bit Rate (BDBR)

TABLE 4. Performance of the proposed algorithm.

	Sequence	[16]		[17]		[18]		[24]		Proposed	
		BDBR (%)	TS (%)	BDBR (%)	TS (%)	BDBR (%)	TS (%)	BDBR (%)	TS (%)	BDBR (%)	TS (%)
A1	Tango2	1.26	69.45	/	/	1.35	35.29	1.74	51.43	1.52	58.22
	FoodMarket4	/	/	/	/	1.30	41.76	1.24	47.32	1.71	55.43
	Campfire	1.87	51.08	1.02	34.80	0.85	44.33	1.37	56.37	1.64	58.79
	Drums	0.50	53.77	/	/	/	/	/	/	1.33	56.61
	ToddlerFountain	0.37	36.89	/	/	/	/	/	/	1.20	57.20
A2	CatRobot	0.85	44.74	1.06	38.71	1.38	39.38	2.04	51.94	1.70	59.04
	DaylightRoad2	0.75	45.58	/	/	0.93	40.56	1.42	58.61	1.55	54.17
	ParkRunning3	/	/	/	/	0.60	42.33	1.07	55.93	1.16	58.07
	RollerCoaster	1.39	56.48	/	/	/	/	/	/	1.82	57.27
	TrafficFlow	1.21	53.41	/	/	/	/	/	/	1.49	56.39
B	ParkScene	0.35	32.09	/	/	0.96	42.28	1.45	58.73	1.29	60.07
	BasketballDrive	0.55	55.09	/	/	1.61	40.32	1.76	53.34	1.40	59.21
	BQTerrace	0.55	33.12	0.97	33.89	1.06	41.00	1.23	55.37	1.58	61.54
	Cactus	0.63	37.53	1.05	35.86	1.27	41.31	2.07	51.37	1.41	53.75
	Kimono	0.41	53.00	1.13	38.56	1.13	41.15	1.31	54.68	1.26	54.38
	MarketPlace	/	/	0.87	34.12	/	/	/	/	1.58	57.24
C	BasketballDrill	1.05	29.90	1.25	38.40	1.86	40.85	1.60	53.92	1.88	59.07
	PartyScene	0.13	28.03	1.16	34.83	1.00	42.17	0.87	52.74	1.32	60.02
	RaceHorsesC	0.63	32.28	0.89	37.69	0.71	42.03	1.25	51.07	1.78	52.37
	BQMall	0.58	30.77	/	/	1.27	40.01	1.87	53.39	1.81	55.20
D	BlowingBubbles	0.12	22.08	1.08	34.75	0.95	38.34	1.57	51.33	1.46	49.69
	BasketballPass	0.52	27.03	/	/	1.53	40.46	1.53	53.96	1.73	62.34
	BQSquare	0.21	19.17	0.94	38.93	0.72	40.19	0.93	54.78	1.33	58.81
	RaceHorses	0.09	23.59	1.34	36.03	0.73	40.61	1.14	55.79	1.17	59.94
E	FourPeople	0.86	27.99	1.05	37.66	1.65	39.73	2.19	55.21	1.62	57.11
	KristenAndSara	0.86	35.70	0.97	37.62	0.42	38.66	1.88	55.46	2.07	56.53
	Johnny	1.06	41.05	/	/	1.80	39.80	2.37	56.43	1.50	57.94
	Average	0.63	39.16	1.06	36.56	1.14	40.57	1.54	54.05	1.53	57.27

to measure the encoding quality; TS to measure the complexity reduction performance. The calculation formula of TS is as follows:

$$TS = \frac{1}{4}\sum_{i=1}^{4}\frac{T_{base}(i) - T_{prop}(i)}{T_{base}(i)} \times 100\% \qquad (13)$$

where $\boldsymbol{T_{base(i)}}$ and $\boldsymbol{T_{pro(i)}}$ respectively indicate the encoding time used by the original VTM10.0 and the encoding time used by the rapid algorithm suggested in this article under the QP values of 22, 27, 32, and 37, respectively.

B. COMPARISON AND ANALYSIS OF EXPERIMENTAL RESULTS

Table 4 shows that our algorithm can reduce the encoding time by 57.27% compared with the anchor algorithm. At the same time, the BDBR of the algorithm is only increased by 1.53%, which is acceptable. Our test sequences contain videos of different resolutions, which demonstrates the good generalization ability of the proposed method. Among these, it is worth noting that for a single sequence, the proposed method achieves a remarkable maximum reduction of 61.54% in encoding time (observed in BQTerrace).

In order to verify the effectiveness of the proposed algorithm, we compared the algorithm proposed in this paper with the advanced algorithms described in literature [16], [17], [18], [24]. These experiments are performed on the same sequence, configuration (AI) and parameters to ensure a fair and accurate comparison. Reference [16] also uses the methods of CNN and RFC at the same time. In [16], CNN is used to judge the depth of CU, while the CNN in our joint method is to judge whether the CU is divided. Reference [17] combines traditional methods and CNN methods, CNN and traditional methods are widely used in the field of video coding. Reference [18] combines visual perception methods and RFC methods. It can be seen that the methods used in [16], [17], and [18] are related to the algorithms we proposed, and they include data-driven methods and heuristic methods, which are representative to a certain extent. It is worth noting that [17] and [24] are related to our previous work. To ensure a fair and accurate comparison, these experiments need to be performed on the same sequence, configuration (AI) and parameters.

By observing Table 5, it can be found that our proposed algorithm reduces the encoding time by 18.16%, 20.46%,

TABLE 5. Performance comparison of proposed algorithms.

Sequence		The performance of [16] on different class sequences（Average）		The performance of the proposed algorithm on different class sequences（Average）	
		BDBR (%)	TS (%)	BDBR (%)	TS (%)
Sequences used in Table 6 of [16]	Class A (8 sequences)	1.03	51.43	1.53	57.21
	Class B (5 sequences)	0.50	42.17	1.39	57.79
	Class C (4 sequences)	0.60	30.25	1.70	56.67
	Class D (4 sequences)	0.24	22.97	1.42	57.70
	Class E (3 sequences)	0.93	34.91	1.73	57.19
(Under the above 24 sequences)		0.63	39.16	1.54	57.32
Sequence		**The performance of [17] on different class sequences（Average）**			
		BDBR (%)	**TS (%)**	**BDBR (%)**	**TS (%)**
Sequences used in Table 3 of [17]	Class A (2 sequences)	1.04	36.76	1.67	58.92
	Class B (4 sequences)	1.01	35.61	1.46	56.73
	Class C (3 sequences)	1.10	36.97	1.66	57.15
	Class D (3 sequences)	1.12	36.57	1.32	56.15
	Class E (2 sequences)	1.01	37.64	1.85	56.82
(Under the above 14 sequences)		1.06	36.56	1.56	57.02
Sequence		**The performance of [18] on different class sequences（Average）**			
		BDBR (%)	**TS (%)**	**BDBR (%)**	**TS (%)**
Sequences used in Table 2 of [18]	Class A (6 sequences)	1.07	40.61	1.55	57.29
	Class B (5 sequences)	1.21	41.21	1.39	57.79
	Class C (4 sequences)	1.21	41.27	1.70	56.67
	Class D (4 sequences)	0.98	39.90	1.42	57.70
	Class E (3 sequences)	1.29	39.40	1.73	57.19
(Under the above 22 sequences)		1.14	40.57	1.54	57.35
Sequence		**The performance of [24] on different class sequences（Average）**			
		BDBR (%)	**TS (%)**	**BDBR (%)**	**TS (%)**
Sequences used in Table 1 of [24]	Class A (6 sequences)	1.48	53.60	1.55	57.29
	Class B (5 sequences)	1.56	54.70	1.39	57.79
	Class C (4 sequences)	1.40	52.78	1.70	56.67
	Class D (4 sequences)	1.29	53.97	1.42	57.70
	Class E (3 sequences)	2.15	55.70	1.73	57.19
(Under the above 22 sequences)		1.54	54.05	1.54	57.35

16.78, and 3.3% on average compared with the algorithms described in [16], [17], [18], and [24].

The proposed algorithm in this paper shows comparable performance to the algorithm of [16] in the Class A high-resolution category. Compared with [16], the algorithm discussed in this paper shows a slight increase (in class A, the average encoding time is reduced by 5.78% compared with [16]). But the algorithm discussed in this paper shows superior performance in low-resolution classes (i.e., Class C and Class D) (in Class D, the average encoding time is reduced by34.73% compared to [16]).

Compared with [17], our proposed algorithm has a good effect in high-resolution sequences, and the proposed algorithm is 22.16% higher than [17] on average in class A sequences. Our algorithm also shows a slight increase in low-resolution sequences, which is 19.58% higher in class D sequences. In Figure 7, we selected several sequences and compared with [17] using the time saved as the evaluation metric.

Compared to [18] and [24], we not only save 16.78% and 3.3% of time, but also control the increase of BDBR.

In addition, we encoded the sequence BasketballDrill with the algorithm proposed in this chapter and the original algorithm of VTM10.0 respectively, and randomly selected a frame to compare the CU division of the two, as shown in Figure 8. It can be seen that whether in the texture complex region or in the texture smooth region, the proposed algorithm exhibits mostly the same CU division as the original algorithm, which also proves that the proposed algorithm can basically replace the original algorithm to complete the

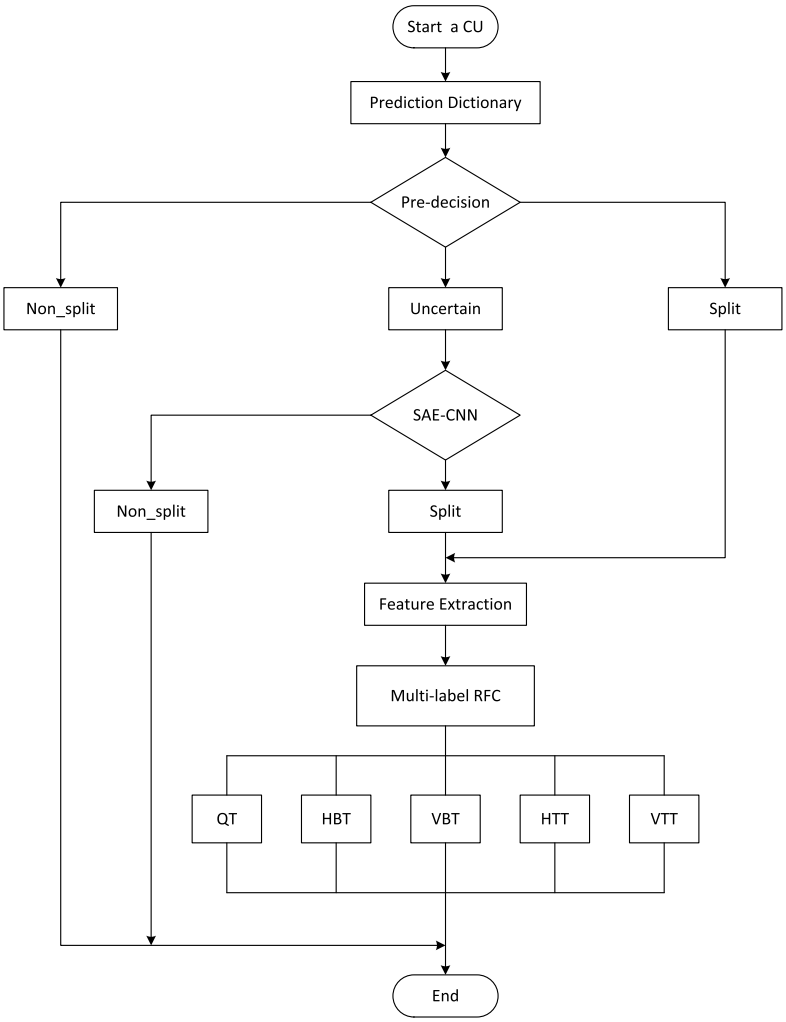


FIGURE 6. Overall algorithm flow chart.

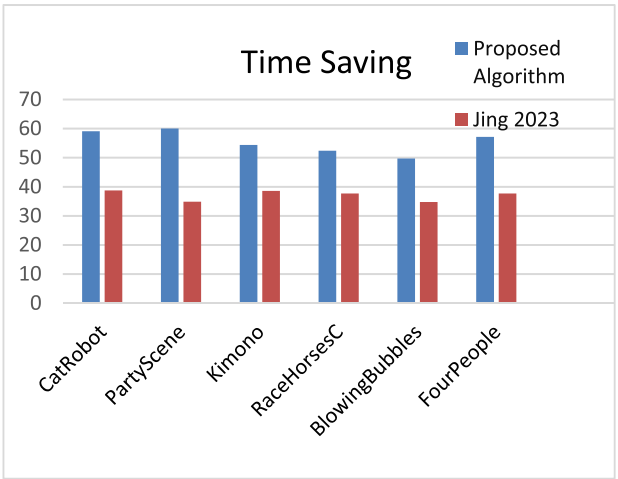
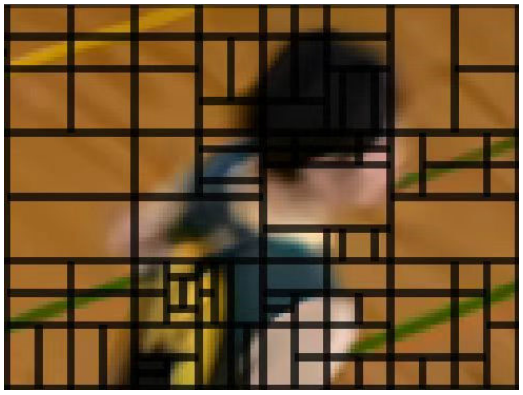


FIGURE 7. Histogram comparing the experimental results of the proposed algorithm with those of ref. [17].

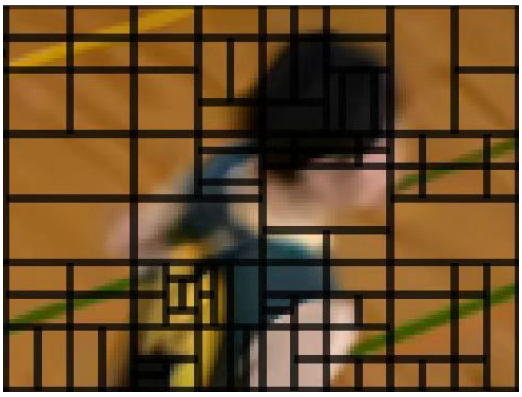
CU division, and the division results are close to the original algorithm's optimal division results.

In order to gain a clearer insight into the rate-distortion (RD) performance of the proposed algorithm, we select BasketballDrill and BQSquare as illustrative examples and perform coding experiments using both the VTM10.0 anchor algorithm and the algorithm proposed in this paper. As shown in Figure 9, it can be obtained that the RD curves of the proposed algorithm and the VTM10.0 anchor algorithm highly overlap with the proposed algorithm, which means that the proposed method and the anchor algorithm have similar RD performance.

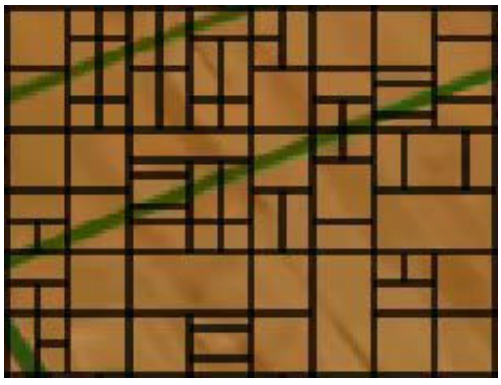
In our previous work [8], a VVC complexity reduction algorithm based on statistical analysis and adaptive size convolutional neural network is proposed to make partition decisions for CUs of different sizes. However, the previous work can only make a decision on whether to divide a CU, and the specific division type decision still needs to use the RDO process of the traditional algorithm, which needs further improvement. Aiming at this shortcoming, this paper designs an RFC-based CU partition mode decision algorithm based on [8], and combines the previous work, proposes a fast adaptive CU partition decision algorithm within a VVC frame. So as to certify the innovation and effectiveness of the decision algorithm of CU partition type based on JRFC suggested in this article, we compared the previous work



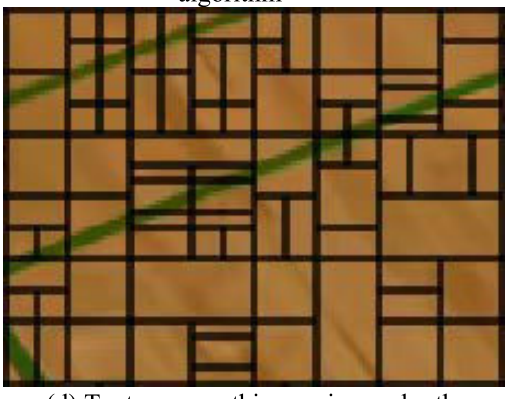
(a) Texture complex areas under VTM 10.0



(b) Texture complex region under the proposed algorithm



(c) Texture smoothing area under VTM 10.0



(d) Texture smoothing region under the proposed algorithm

FIGURE 8. Comparison of CU division for different texture complexity.

TABLE 6. Experimental result table of ablation study.

Sequence		[8]		Proposed	
		BDBR (%)	TS (%)	BDBR (%)	TS (%)
A1	Campfire	1.01	35.78	1.64	58.79
A2	CatRobot	0.96	36.98	1.70	59.04
B	BQTerrace	0.89	36.79	1.58	61.54
	Cactus	0.87	34.12	1.41	53.75
	Kimono	0.71	34.59	1.26	54.38
C	BasketballDrill	1.10	35.03	1.88	59.07
	PartyScene	0.67	34.55	1.32	60.02
	RaceHorsesC	0.75	33.89	1.78	52.37
D	BlowingBubbles	0.97	35.86	1.46	49.69
	BQSquare	0.71	32.35	1.33	58.81
	RaceHorses	0.76	33.14	1.17	59.94
E	FourPeople	0.99	38.40	1.62	57.11
	KristenAndSara	1.08	35.84	2.07	56.53
	Average	0.88	35.17	1.56	57.00

[8] to conduct ablation experiments. For the specific experimental results, please refer to Table 6. It can be seen from the experimental results that the CU partition type decision algorithm suggested in this article reduces the complexity

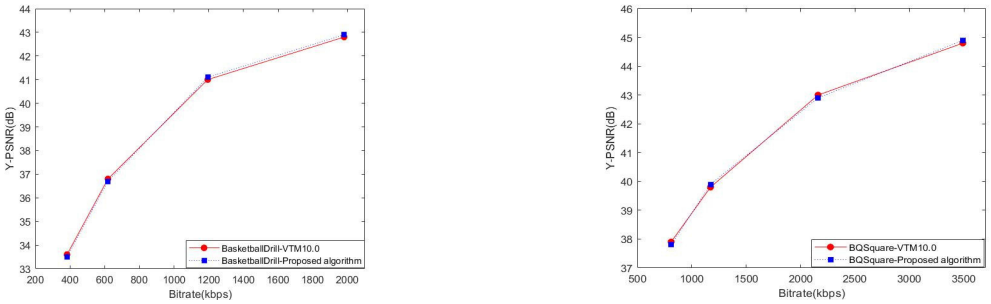


FIGURE 9. The RD performance graph of the proposed algorithm. (a) RD of the BasketballDrill; (b) RD of the BQSquare.

by an average of 21.83% compared with [8]. It also proves that the suggested algorithm has excellent performance of reducing complexity.

V. CONCLUSION

In this article, we first put forward a decision algorithm of CU partition type based on JRFC, using 5 independent binary classifiers to design JRFC. And through experiments and theoretical analysis, the appropriate features are selected as the input of JRFC, which is used to make decisions on the division type of CUs of different sizes. The results of the ablation experiments can reflect that the decision algorithm of CU partition type based on JRFC is innovative and effective, and can effectively reduce the coding complexity. Then, combined with previous work, we put forward fast adaptive CU partition decision algorithm for VVC intra coding. The algorithm can deal with the CUs of different sizes generated by the QTMT structure, and makes partition decisions for CUs with sizes below 32×32, and thus greatly reduce the coding complexity. The experimental findings demonstrate that the algorithm proposed in this article yields a remarkable average reduction of 57.27% in encoding time, while incurring a mere 1.53% increase in BDBR. It has been demonstrated that the proposed algorithm can significantly diminish the intricacy of coding while ensuring virtually unaltered coding quality, thus achieving a commendable equilibrium between the reduction of coding complexity and the compromise of coding quality.

REFERENCES

[1] G. Tang, Y. Hu, H. Xiao, L. Zheng, X. She, and N. Qin, "Design of real-time video transmission system based on 5G network," in *Proc. IEEE 16th Conf. Ind. Electron. Appl. (ICIEA)*, Chengdu, China, Aug. 2021, pp. 522–526, doi: 10.1109/ICIEA51954.2021.9516191.

[2] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)," *Proc. IEEE*, vol. 109, no. 9, pp. 1463–1493, Sep. 2021, doi: 10.1109/JPROC.2020.3043399.

[3] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012, doi: 10.1109/TCSVT.2012.2221192.

[4] A. Wieckowski, J. Ma, H. Schwarz, D. Marpe, and T. Wiegand, "Fast partitioning decision strategies for the upcoming versatile video coding (VVC) standard," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 4130–4134, doi: 10.1109/ICIP.2019.8803533.

[5] S. Bouaafia, R. Khemiri, and F. E. Sayadi, "Rate-distortion performance comparison: VVC vs. HEVC," in *Proc. 18th Int. Multi-Conference Syst., Signals Devices (SSD)*, Monastir, Tunisia, Mar. 2021, pp. 440–444, doi: 10.1109/SSD52085.2021.9429377.

[6] Y. Fan, J. Chen, H. Sun, J. Katto, and M. Jing, "A fast QTMT partition decision strategy for VVC intra prediction," *IEEE Access*, vol. 8, pp. 107900–107911, 2020, doi: 10.1109/ACCESS.2020.3000565.

[7] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block partitioning structure in the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3818–3833, Oct. 2021.

[8] J. Zhao, P. Dai, and Q. Zhang, "A complexity reduction method for VVC intra prediction based on statistical analysis and SAE-CNN," *Electronics*, vol. 10, no. 24, p. 3112, Dec. 2021, doi: 10.3390/electronics10243112.

[9] G. Zhang, L. Xiong, X. Lian, and W. Zhou, "A CNN-based coding unit partition in HEVC for video processing," in *Proc. IEEE Int. Conf. Real-Time Comput. Robotics (RCAR)*, Irkutsk, Russia, Aug. 2019, pp. 273–276, doi: 10.1109/RCAR47638.2019.9043972.

[10] C. Xu, P. Liu, Y. Wu, K. Jia, and W. Dong, "A fast CTU depth selection algorithm for H.265/HEVC based on machine learning," in *Proc. IEEE 3rd Int. Conf. Signal Image Process. (ICSIP)*, Shenzhen, Jul. 2018, pp. 154–161, doi: 10.1109/SIPROCESS.2018.8600458.

[11] H.-Y. Hsu, S.-E. Huang, and Y. Lin, "Computational complexity reduction for HEVC intra prediction with SVM," in *Proc. IEEE 6th Global Conf. Consum. Electron. (GCCE)*, Nagoya, Oct. 2017, pp. 1–2, doi: 10.1109/GCCE.2017.8229195.

[12] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2225–2238, Jul. 2015, doi: 10.1109/TIP.2015.2417498.

[13] M. Zhang, X. Zhai, Z. Liu, and C. An, "Fast algorithm for HEVC intra prediction based on adaptive mode decision and early termination of CU partition," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, Mar. 2018, p. 434, doi: 10.1109/DCC.2018.00087.

[14] A. Tissier, W. Hamidouche, J. Vanne, F. Galpin, and D. Menard, "CNN oriented complexity reduction Of VVC intra encoder," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3139–3143, doi: 10.1109/ICIP40778.2020.9190797.

[15] T. Li, M. Xu, R. Tang, Y. Chen, and Q. Xing, "DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC," *IEEE Trans. Image Process.*, vol. 30, pp. 5377–5390, 2021, doi: 10.1109/TIP.2021.3083447.

[16] Y.-H. Huang, J.-J. Chen, and Y.-H. Tsai, "Speed Up H.266/QTMT intra-coding based on predictions of ResNet and random forest classifier," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2021, pp. 1–6, doi: 10.1109/ICCE50685.2021.9427626.

[17] Z. Jing, W. Zhu, and Q. Zhang, "A fast VVC intra prediction based on gradient analysis and multi-feature fusion CNN," *Electronics*, vol. 12, no. 9, p. 1963, Apr. 2023, doi: 10.3390/electronics12091963.

[18] M.-J. Chen, C.-A. Lee, Y.-H. Tsai, C.-M. Yang, C.-H. Yeh, L.-J. Kau, and C.-Y. Chang, "Efficient partition decision based on visual perception and machine learning for H.266/Versatile video coding," *IEEE Access*, vol. 10, pp. 42141–42150, 2022, doi: 10.1109/ACCESS.2022.3168155.

[19] Q. Zhang, Y. Wang, L. Huang, and B. Jiang, "Fast CU partition and intra mode decision method for H.266/VVC," *IEEE Access*, vol. 8, pp. 117539–117550, 2020, doi: 10.1109/ACCESS.2020.3004580.
[20] T. Fu, H. Zhang, F. Mu, and H. Chen, "Fast CU partitioning algorithm for H.266/VVC intra-frame coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, Jul. 2019, pp. 55–60, doi: 10.1109/ICME.2019.00018.
[21] Q. He, W. Wu, L. Luo, C. Zhu, and H. Guo, "Random forest based fast CU partition for VVC intra coding," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Chengdu, China, Aug. 2021, pp. 1–4, doi: 10.1109/BMSB53066.2021.9547117.
[22] T. Amestoy, A. Mercat, W. Hamidouche, C. Bergeron, and D. Menard, "Random forest oriented fast QTBT frame partitioning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 1837–1841, doi: 10.1109/ICASSP.2019.8683413.
[23] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard, and C. Bergeron, "Tunable VVC frame partitioning based on lightweight machine learning," *IEEE Trans. Image Process.*, vol. 29, pp. 1313–1328, 2020, doi: 10.1109/TIP.2019.2938670.
[24] J. Zhao, A. Wu, and Q. Zhang, "SVM-based fast CU partition decision algorithm for VVC intra coding," *Electronics*, vol. 11, no. 14, p. 2147, Jul. 2022, doi: 10.3390/electronics11142147.

LINA SI received the master's degree in computer technology from the Huazhong University of Science and Technology, Wuhan, China, in 2010. She is currently a Professor with the Zhengzhou University of Light Industry, China. Her current research interests include video processing, Bayesian estimation, and intelligent computation.

WENDI ZHU received the B.S. degree in information engineering from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2020, where he is currently pursuing the master's degree in computer technology with the School of Computer and Communication Engineering. His current research interests include image processing, deep learning, versatile video coding, and extensions of the versatile video coding.

QIUWEN ZHANG received the Ph.D. degree in communication and information systems from Shanghai University, Shanghai, China, in 2012. Since 2012, he has been with the College of Computer and Communication Engineering, Zhengzhou University of Light Industry, where he is currently an Associate Professor. He has published over 30 technical articles in the field of pattern recognition and image processing. His major research interests include 3D signal processing, machine learning, pattern recognition, video codec optimization, and multimedia communication.