

Tunable VVC Frame Partitioning Based on Lightweight Machine Learning

Thomas Amestoy¹, Alexandre Mercat, Wassim Hamidouche², Daniel Menard, and Cyril Bergeron

Abstract—Block partition structure is a critical module in video coding scheme to achieve significant gap of compression performance. Under the exploration of the future video coding standard, named Versatile Video Coding (VVC), a new Quad Tree Binary Tree (QTBT) block partition structure has been introduced. In addition to the QT block partitioning defined in High Efficiency Video Coding (HEVC) standard, new horizontal and vertical BT partitions are enabled, which drastically increases the encoding time compared to HEVC. In this paper, we propose a lightweight and tunable QTBT partitioning scheme based on a Machine Learning (ML) approach. The proposed solution uses Random Forest classifiers to determine for each coding block the most probable partition modes. To minimize the encoding loss induced by misclassification, risk intervals for classifier decisions are introduced in the proposed solution. By varying the size of risk intervals, tunable trade-off between encoding complexity reduction and coding loss is achieved. The proposed solution implemented in the JEM-7.0 software offers encoding complexity reductions ranging from 30% to 70% in average for only 0.7% to 3.0% Bjøntegaard Delta Rate (BD-BR) increase in Random Access (RA) coding configuration, with very slight overhead induced by Random Forest. The proposed solution based on Random Forest classifiers is also efficient to reduce the complexity of the Multi-Type Tree (MTT) partitioning scheme under the VTM-5.0 software, with complexity reductions ranging from 25% to 61% in average for only 0.4% to 2.2% BD-BR increase.

Index Terms—Video compression, VVC, JEM, VTM, QTBT, complexity reduction, machine learning, random forest.

I. INTRODUCTION

THE expansion of Internet coupled with the rapid introduction of Ultra High Definition (UHD), High Dynamic Range (HDR) and 360° video contents in daily life have caused the explosion of video traffic. Recent study published

in Cisco [1] has predicted that video traffic will increase from 61% of the global IP traffic in 2016 to 82% in 2021. This increasing demand for video contents brings new challenges to compression, especially to enhance the coding efficiency and enable a high quality of experience of video services. Moreover, in the context of embedded systems with limited computing and energy resources, the complexity of video codecs is a crucial challenge to reach real time processing with low energy consumption.

The ITU-T Video Coding Experts Group and ISO/IEC Moving Picture Expert Group have released in 2013 the High Efficiency Video Coding (HEVC) standard [2]. HEVC reduces the bitrate up to 60% compared to its predecessor Advanced Video Coding (AVC) standard [3] for the same subjective video quality [4], [5]. The Joint Video Exploration Team (JVET) has been recently established to investigate several new coding tools under the Joint Exploration Model (JEM) software [6] [7] in order to show the interest of developing a new video coding standard called Versatile Video Coding (VVC), with coding capability beyond HEVC. These new coding tools already increase the coding efficiency by up to 40% compared to HEVC [8]. However, bitrate savings come with a significant complexity increase compared to HEVC.

At the encoder side, computationally expensive tools have been added in the JEM, especially for intra prediction and frame partitioning scheme that select the appropriate block size according to the local activity of the pixels. The new Quad Tree Binary Tree (QTBT) partitioning scheme introduced in the JEM improves the coding efficiency by approximately 5% [9] in Random Access (RA) configuration. However, the coding gain is achieved at the expense of considerable complexity increase. This complexity becomes a bottleneck for the development of the VVC standard and may interfere with its deployment especially on embedded platforms and live applications.

To reduce the computational complexity of encoders, several techniques propose to reduce the tested intra mode candidates. These techniques use features such as gradients of luminance samples [10]–[12] or Machine Learning (ML) techniques [13] to predict a reduced set of likely intra modes. Other techniques reduce the complexity of the encoding process by focusing on the partitioning scheme and testing a reduced number of partition configurations. In HEVC, to predict the most probable Quad Tree (QT) partitions, these techniques leverage intermediate encoding information [14]–[18], texture

Manuscript received January 24, 2019; revised July 10, 2019 and August 23, 2019; accepted August 27, 2019. Date of publication September 6, 2019; date of current version November 4, 2019. This work was supported in part by the French FUI Project EFIGI and in part by the European Celtic-Plus Project VIRTUOSE. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xin Li. (Corresponding author: Thomas Amestoy.)

T. Amestoy is with HTE/STR/MMP Gennevilliers, Thales SIX GTS France, 92230 Gennevilliers, France, and also with INSA Rennes, Institute of Electronic and Telecommunication of Rennes, CNRS—UMR 6164, 35708 Rennes, France (e-mail: thomas.amestoy@thalesgroup.com).

A. Mercat is with the Department of Computing Sciences, Tampere University, 33720 Tampere, Finland (e-mail: alexandre.mercat@tuni.fi).

W. Hamidouche and D. Menard are with INSA Rennes, Institute of Electronic and Telecommunication of Rennes, CNRS—UMR 6164, 35708 Rennes, France (e-mail: wassim.hamidouche@insa-rennes.fr; daniel.menard@insa-rennes.fr).

C. Bergeron is with HTE/STR/MMP Gennevilliers, Thales SIX GTS France, 92230 Gennevilliers, France (e-mail: cyril.bergeron@thalesgroup.com).

Digital Object Identifier 10.1109/TIP.2019.2938670

1057-7149 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

characteristics [19]–[21], motion divergence [22]–[24] or ML solutions [24]–[28]. More recently, some techniques have already investigated the complexity reduction of QTBT partitioning scheme [29]–[34]. Even though Random Forest (RF) is a classical method in ML that offers high classification performance with slight overhead and is widely used in many applications such as image classification [35] and 3D pose estimation [36], none of the previously mentioned QT or QTBT fast partitioning techniques rely on RF. Furthermore, tunable complexity reduction for QTBT partitioning has not yet been investigated in previous works.

In this paper, we propose a tunable ML solution based on RF classifiers to speed up the QTBT partitioning scheme in RA coding configuration. The proposed solution explores a novel approach to solve a 4 classes classification problem inherent to QTBT partitioning scheme, which has not been fully studied in related works. The decision of QTBT partition mode for the Coding Unit (CU) is modeled as three distinct binary classification problems. Thus, three binary RF classifiers are trained independently off-line, with separate training for each CU size. The goal of the classifiers is to skip expensive exploration of the partition modes classified as unlikely. Furthermore, the classifiers take as input only features from the current CU, making the solution parallel-friendly. To limit the Rate Distortion (RD) loss induced by misclassification, risk intervals are introduced to control the classifier decisions. When the classifier decision falls into the risk interval, all possible partitioning modes are processed. The risk intervals are set by the encoder based on the encoding of a reference frame, adapting the RD loss induced by misclassification to the encoded content. By varying the size of risk intervals, tunable complexity reduction is achieved.

To the best of our knowledge, the proposed solution is the first tunable complexity reduction solution applied on an encoder post HEVC. It includes various Complexity Reduction Configurations (CRCs), each offering a new trade-off between complexity reduction and Bjøntegaard Delta Rate (BD-BR) increase. In JEM-7.0 software, encoding complexity reductions vary from 30% to 70% in average at the expense of only 0.7% to 3.0% BD-BR increase. The proposed solution based on RF classifiers is also efficient to reduce the complexity of the Multi-Type Tree (MTT) partitioning scheme in the VVC Test Model (VTM)-5.0 software, with complexity reductions varying from 25% to 61% in average for limited BD-BR increase of 0.4% to 2.2%. Moreover, the proposed solution induces a very low overhead between 0.2% and 1.8% of the encoding time according to the video content, which is a key point to adopt this solution in a real-time and embedded framework.

The rest of the paper is organized as follows. Section II describes the frame partitioning decision in HEVC and JEM, and then reviews the related works. Section III goes through background of RF classifiers and presents the proposed classification problem. Section IV depicts the training dataset. The training process to build RF classifiers is described in Section V. Section VI details how tunable encoding complexity reduction is achieved using various configurations of the risk intervals. Experimental results are presented and

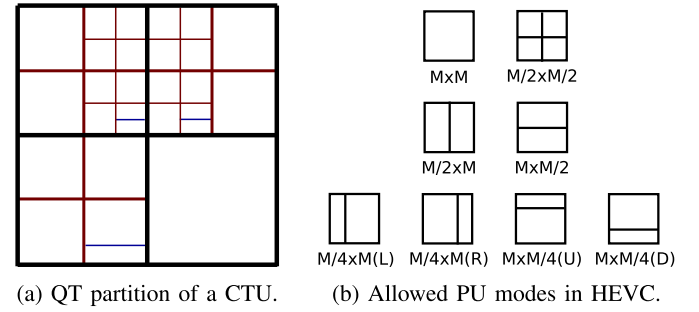


Fig. 1. QT partition scheme of a CTU in HEVC. QT partition modes in red and further PU modes in blue.

analyzed for both JEM-7.0 and VTM-5.0 in Section VII. Finally, Section VIII concludes this paper.

II. RELATED WORKS

A. Overview of Frame Partitioning Scheme in HEVC and JEM

HEVC relies on the classic hybrid video coding combining intra/inter image predictions and transform coding. Intra prediction exploits spatial redundancy within the same frame whereas Inter prediction exploits temporal redundancy in the video sequence. First, each frame is split into Coding Tree Units (CTUs) of equal sizes. As shown in Fig. 1a, each CTU is then recursively split into square CUs, following a QT partitioning scheme. To perform the prediction, the CUs can be divided into Prediction Units (PUs) of smaller size following one of the eight PU modes illustrated in Fig. 1b.

After performing prediction, residual blocks can be further split recursively with a second QT partitioning scheme into Transform Units (TUs) on which transform is performed. However, the QT scheme suffers from the following limitations:

- CUs can only be square, with no flexible shape to cover all block characteristics.
- Luma and Chroma components of the encoded sequence have the same QT splitting which is rarely optimal for chroma.
- Residual can only be split into TUs with square shapes, reducing the potential impact of transformation.

To overcome these limitations, a QTBT partitioning scheme has been proposed in the JEM [37]. QTBT is an extension to the QT partitioning that enables symmetric binary partition modes CUs in both horizontal (Binary Tree Horizontal (BTH)) and vertical (Binary Tree Vertical (BTV)) directions, as shown in Fig. 2a. When Binary Tree (BT) partition mode is used in a CU, QT partition mode is no longer allowed. In the JVET Common Test Conditions (CTC) [38], BT_{depth} parameter is set to 3, enabling only 3 successive BT partitions. Fig. 2b shows the tree representation of QTBT partition, where the leaf surrounded by a green square corresponds to the green CU of Fig. 2a.

The flexibility brought by BT partition modes covers the PU modes. In JEM, CUs are directly used to perform prediction. Rectangular transforms have also been adopted so that the transform can be applied on the CUs without any further splitting of residual blocks [39]. Moreover, different partition

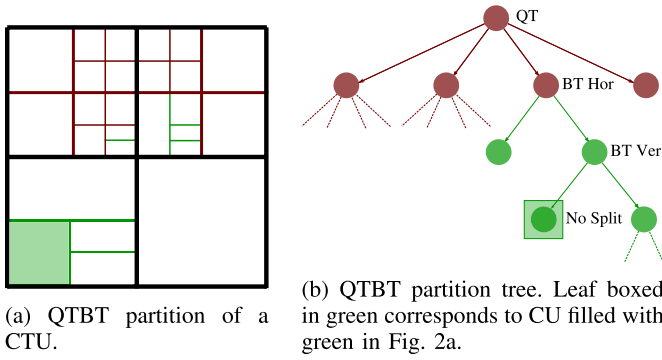


Fig. 2. QTBT partition scheme in JEM. In red QT partition mode and in green BT partition modes.

trees can be applied on Luma and Chroma components for intra predicted slices. For inter predicted slices, the partition trees of Luma are used for Chroma.

B. Complexity Reduction of Frame Partitioning

QTBT partitioning scheme has not yet been fully studied from the perspective of complexity reduction. Reason why in this section, the techniques proposed to speed up the QT partitioning scheme in HEVC are first described. Subsequently, papers that have investigated the complexity reduction of QTBT partitioning scheme are presented.

1) *Partitioning Scheme in HEVC*: the techniques proposed to speed up the QT partitioning scheme in HEVC are divided into four categories, whether they involve: intermediate encoding information, texture characteristics, motion divergence, and ML.

a) *Intermediate encoding information*: techniques of the first category are based on intermediate information computed during the encoding process such as depths of previously encoded blocks, encoding flags or RD cost of PU modes. Pan *et al.* [14] use the motion estimation and all-zero block detection informations of $M \times M$ PU mode (Fig. 1a), coupled with merge mode informations of previous depths to determine if early merge can save computational time. A threshold based on Sum of Absolute Transform Differences (SATD) of $M \times M$ PU mode has been proposed in [15]. The technique ignores the unlikely remaining PU modes or early terminating QT partitioning mode. In [16], Correa *et al.* figure out that some QT depths tend to be used in co-located areas of adjacent frames and exploit this correlation to target a computational complexity reduction. Same authors adapt this principle to speedup the coding of motion sequences in [17] by using a motion compensated area as reference in previously encoded frame (instead of co-located area) and by using depths of spatial neighboring CTUs. In [18], a thresholding process applied on RD cost of $M \times M$ merge mode determines if the skip mode shall be used for the PU. These techniques [15] [17] [18] reduce the encoding complexity between 30% and 50% with a BD-BR increase in the range 0.4% to 1.4%.

b) *Texture characteristics*: second category covers techniques exploiting the correlations between spatial information of a sequence, also called texture, and the QT partitioning scheme. Usually, areas having complex textures are split into

small blocks in order to fit better its local variations. Authors in [19] apply thresholds on local and global edge filters of luminance samples in 4 directions ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) to determine if a CU shall be split, non-split or undetermined. A similar principle is applied in [20], where the authors compute adaptive thresholds based on texture homogeneity for early determination of QT depth of CTUs. Mercat *et al.* propose in [21] a technique that determines at each step if the 4 smaller CUs shall be merged into one CU. This bottom-up approach exploits correlation between the QT partitioning scheme and texture variance.

c) *Motion divergence*: techniques of the third category explore the correlation between motion divergence in a frame and the QT partitioning scheme. Areas of the frame with continuous motion tend to be split into larger blocks. Reason why in [22], authors detect motion continuity by applying Sobel operators to a pre-computed optical flow and use this information to predict block size and save computational time. In [23], a score called Pyramid Motion Divergence (PMD) based on variance of Motion Divergence Field is assigned to every CU. The authors show that CUs with similar PMD tend to be encoded with the same partitioning. Blasi *et al.* [24] propose a bottom-up approach to ignore certain CU partition modes based on a motion vector variance distance, computed on the four QT sub-CUs.

d) *Machine Learning*: techniques of last category use ML approaches to reduce encoding complexity. They either take as input features extracted from one or more of previous categories, or rely on Deep Learning (DL) approaches such as Convolution Neural Networks (CNNs). Liu *et al.* [25] propose a technique based on CNN for All Intra (AI) configuration that determine if a CU must be early terminated or early split. A CNN is separately trained for every QT CU size from 32×32 to 8×8 pixels. In [26], Duanmu *et al.* focus on Screen Coding Content (SCC), an extension of HEVC that targets typical screen contents. Separate CNN classifiers are separately trained for different Quantization Parameter (QP) values and CU sizes to output a variable between 0 and 1, representing the probability that a CU is early terminated. In [27], to reduce RA encoding complexity in HEVC, Support Vector Machines (SVM) classifiers are separately trained for every CU size in order to determine if a CU should be early terminated. The SVM classifier takes as an input features such as pixel gradients, sub-CUs Movement Vectors (MVs) and intermediate encoding information (encoding flags, CU depth of neighboring blocks and the RD cost). The technique proposed by Blasi *et al.* [24] is based on decision trees, each relying on spatial features and intermediate RA configuration encoding information. The decision trees are trained separately for every CU size from 64×64 to 8×8 . In [28], Shen *et al.* propose a Bayesian rule classifier employing features of $M \times M$ PU mode such as SATD, RD cost and MV to determine the PU mode of a CU. Finally, Mercat *et al.* [40] compares features used in state of the art techniques to predict HEVC partition, considering both information gain and computational complexity.

2) *QTBT Partitioning Scheme*: QTBT partitioning scheme in the JEM allows two more partition modes compared

to QT partitioning scheme, considerably increasing the encoding complexity. Furthermore, intermediate PU splitting modes, that provides useful information for early termination or early skip decision in HEVC, have been removed in the JEM. For these reasons, most of previously mentioned techniques can not be directly used to speedup the JEM encoder.

In [29], Yamamoto proposes to reduce the encoding complexity in RA configuration by setting high value of BT_{depth} on frames with low Temporal ID, whereas frames with high Temporal ID use smaller value of BT_{depth} . In the QTBT partition scheme the same CU can be generated by different block partition choices. Huang *et al.* [30] reduce the encoding complexity by re-using the encoder decisions of the same CU explored in previous partition choices. The technique proposed by Lin *et al.* [31] skips the BT Rate Distortion Optimization (RDO) process of the second sub-CU, when the RD cost of the parent CU and the first sub CU fulfil certain constraints. Authors in [32] and [33], use CNNs to predict a depth description of QTBT partition of the CTUs. In [32], the CNN takes as an input the 32×32 pixels blocks of the frame, as well as QP value, and outputs a class from 0 to 5 describing QTBT partition depth for AI configuration. In [33], the false prediction risk of CNN is controlled based on temporal correlation for RA configuration. Wang *et al.* [34] use a combination between Motion Divergence Field and gradient of luminance samples to model the RD cost of a CU. A probabilistic model is then proposed to determine unlikely partition modes of the QTBT partitioning scheme. Complexity reduction techniques [29], [33] and [34] can reduce encoding complexity in RA configuration by 17%, 32% and 52% for 0.5%, 0.5% and 1.4% of BD-BR increase in average, respectively. These results depend on the encoder version of the used software, encoding parameters and hardware configuration. They nevertheless provide an order of magnitude of the techniques efficiencies.

Unlike previously mentioned techniques speeding up the QTBT partition scheme, the proposed solution is tunable and offers various complexity reduction opportunities, from 30% to 70% in average. Moreover, this solution can be considered as lightweight since it relies on RF classifiers, inducing much smaller overhead compared to CNN based techniques [32] [33].

III. RANDOM FORESTS FOR PARTITION DECISION CLASSIFICATION PROBLEM

This section introduces RF classifiers and presents the three binary classification problems proposed in this work to reduce the number of processed partition modes.

A. Background for Random Forests

Classification by RF [41] is a classical method in ML. RF classifiers predict the value of a target variable, named class, from values of several input variables, named features. They bag many single little-correlated decision trees and gather the results from all the trees to make the final decision.

1) *Build Decision Trees*: Decision trees are constructed by a recursive partitioning of the data set into subsets called nodes. At each node, a threshold that achieves optimal separation of the classes is selected among the input features. Each child-node corresponds to a set of values of the selected input features, so that the totality of child-nodes cover all possible values of this input features.

The criterion used in this work to select the best split for a node in decision trees is Mutual Information (MI). The differential entropy \mathbb{H} of a continuous random variable X is computed as follows:

$$\mathbb{H}(X) = - \int_{\mathbb{R}} f_X(x) \log_2(f_X(x)) dx, \quad (1)$$

where $f_X : \mathbb{R} \rightarrow [0, +\infty]$ is the Probability Density Function (Pdf) of X . The entropy \mathbb{H} measures the quantity of information delivered by the knowledge of X . MI of class C and feature F , noted $\mathbb{I}(F, C)$, is defined as the entropy decreasing of C when F is known [42]. The value of $\mathbb{I}(F, C)$ is comprised between 0 and $\mathbb{H}(C)$, and is expressed by Equation (2)

$$\mathbb{I}(F, C) = \mathbb{H}(C) - \mathbb{H}(C|F). \quad (2)$$

In other words, $\mathbb{I}(F, C)$ measures the information shared by C and F . Therefore, the higher $\mathbb{I}(F, C)$, the more the feature F is relevant to estimate class C . When MI is used as a criterion, the optimal threshold of a feature F to split a decision tree node is the threshold that maximizes $\mathbb{I}(F, C)$ on both subsets of child-nodes.

2) *Benefits of Random Forests*: Let the error rate be the percentage of wrong classification on the training dataset. By de-correlating trees, RF classifiers achieve a better trade-off between error rate and training data over-fitting compared to a single decision tree classifier.

In order to de-correlate the decision trees of the RF, a random subset of the training dataset is selected to build each decision tree. The decision trees in the RF take as an input all the features. However, the splitting threshold of each node in the decision tree is selected among a random subset of the input features. This random selection decreases the probability that two decision trees in the RF select the same set of features in the same order, and therefore is crucial to de-correlate one decision tree from another.

The following example illustrates the interest of RF classifiers in term of error rate compared to single decision tree classifiers. Assume the RF classifier is composed of 10 perfectly de-correlated decision trees, each with error rate ϵ_i of 0.3: $\epsilon_i = \epsilon = 0.3, \forall i \in \{1, \dots, 10\}$.

If a RF classifier makes a wrong prediction when more than half of the base decision tree classifiers are wrong, the error rate ϵ_{RF} of the RF classifier is computed by Equation (3)

$$\epsilon_{RF} = \sum_{j=6}^{10} \binom{10}{j} \epsilon^j (1 - \epsilon)^{10-j} \approx 0.05. \quad (3)$$

It is not possible to de-correlate perfectly the decision trees of the RF as they are trained on the same data. Nonetheless, the more the decision trees are de-correlated, the closer the error rate of the RF classifier is to ϵ_{RF} .

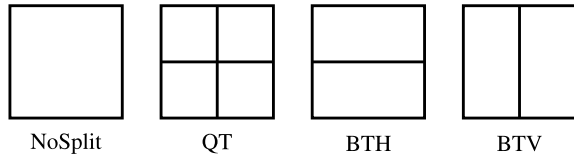


Fig. 3. Four classes of the CU classification problem.

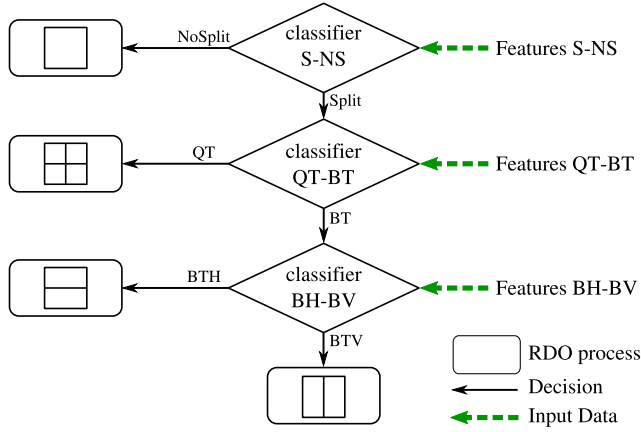


Fig. 4. Convert Four Classes Problem into Three Binary Problems

B. Classification Problem

To find the CU partition mode that achieves the best RD performance, the encoder recursively explores all possible partition modes. This process is called full or exhaustive RDO search. For each CU, the encoder computes the RD cost of the whole CU, QT and BT partition modes, BT partition modes being composed of BTH and BTV partition modes. The encoder selects the partition mode that minimizes the RD cost J , expressed as a trade-off between distortion D and rate R , with λ the Lagrangian multiplier:

$$J = D + \lambda R. \quad (4)$$

The aim of the proposed solution is to predict for every encountered CU the partition mode that minimizes the RD cost. Several partition modes are ignored reducing the number of processed partition modes. As shown in Fig. 3, the problem to solve is a four classes classification problem including the following partition modes: NoSplit, QT, BTH and BTV.

In this work, instead of creating a RF classifier that solves directly the four classes classification problem, the partition decision is divided into three successive binary classification problems. This division adds flexibility into the decision structure and allows a separate training of the classifiers on specific features, improving global classification performance.

The three binary classifiers are named *S-NS*, *QT-BT* and *BH-BV*. As shown in Fig. 4, each classifier takes as input a different set of features and the classifiers are used in the following order:

- 1) **Classifier S-NS**: The two output classes of classifier *S-NS* are either NoSplit partition mode or Split partition modes, where Split partition modes include QT, BTH and BTV partition modes. When the Classifier *S-NS* outputs class NoSplit, NoSplit partition mode is

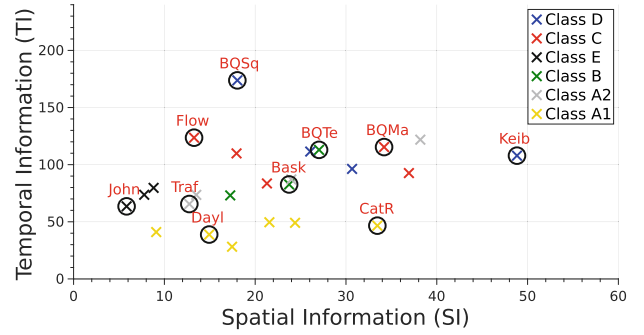


Fig. 5. SI and TI of the CTC sequences according to the classes.

processed and Split partition modes are ignored. Otherwise, NoSplit partition mode is ignored, and the second classifier *QT-BT* is requested.

- 2) **Classifier QT-BT**: The output classes of classifier *QT-BT* are either QT partition mode or BT partition modes, where BT partition modes include BTH and BTV partition modes. When the classifier *QT-BT* outputs QT partition mode, QT partition mode is processed and BT partition modes are ignored. Otherwise, QT partition mode is ignored, and the third classifier *BH-BV* is requested.
- 3) **Classifier BH-BV**: The output classes of classifier *BH-BV* are either BTH partition mode or BTV partition mode. When the classifier *BH-BV* outputs BTH partition mode, BTH partition mode is processed and BTV partition mode is ignored. Otherwise, BTV partition mode is processed and BTH partition mode is ignored.

IV. DATASET CREATION

Let us define a training instance as the entity composed of the chosen set of input features and the associated output class. This section details the creation of the training dataset, i.e. the set that contains all the training instances used to train the RF classifiers.

A. Training Setup

The effectiveness of ML is highly linked to the diversity and relevance of the training dataset. To characterize a video content, Spatial Information (SI) and Temporal Information (TI) metrics are used [43]. The SI estimates the amount of spatial details whereas the TI measures the quantity of motion in the sequence. In Fig. 5, 25 sequences extracted from JVET CTC [38] are represented under the SI TI coordinates.

To cover a wide range of these two content types, the training dataset is extracted from 10 training sequences spanning a large range of SI and TI space and distributed across 6 classes (A1, A2, B, C, D, E). The training sequences are circled in black in Fig. 5 including: *DaylightRoad* and *CatRobot1* (class A1), *Traffic* (class A2), *BasketballDrive* and *BQTerrace* (class B), *Flowervase* and *BQMall* (class C), *BQSquare* and *Keiba* (class D), *Johnny* (class E). The training instances are extracted from encodings carried-out with the JEM-7.0 in RA configuration across the 4 QP values used in CTC: 22, 27, 32 and 37. The corresponding output class of a CU is defined

TABLE I
WIDTH AND HEIGHT OF THE CUs COMPOSING
THE SIZE CATEGORIES

S_0	S_1	S_2	S_3	S_4	S_5	S_6
128x128	128x64	64x64	64x32	32x32	32x16	16x16
	64x128	128x32	32x64	64x16	16x32	32x8
		32x128	128x16	16x64	64x8	8x32
			16x128	128x8	8x64	
				8x128		

as the optimal partition mode selected after a exhaustive RDO process.

Let the size category of a CU depend only on the number of pixels in the CU. TABLE I gives the dimensions in width and height of the CUs composing the 7 size categories. The category S_0 corresponds to the largest CUs and the category S_6 to the smallest CUs. For each CU size category, a separate training dataset is created and a separate RF classifier is trained. With this separation, the features are computed on the same number of pixels in each training dataset.

Sequences with high resolution and high frame rate provide more CTUs to the training datasets compared to the low resolution and low frame rate sequences. To avoid being biased by these particular training sequences, the datasets used for training are composed of a fixed number of CTUs by training sequence. To ensure a fixed number of CTUs by training sequence, the number of frames used for training in a sequence differs according to the sequence class: 7 frames for class A1, 13 frames for class A2, 25 frames for class B, 55 frames for class E, 125 frames for class C and 500 frames for class D. To avoid temporal bias, the CTUs come from frames evenly distributed in their sequence time-line.

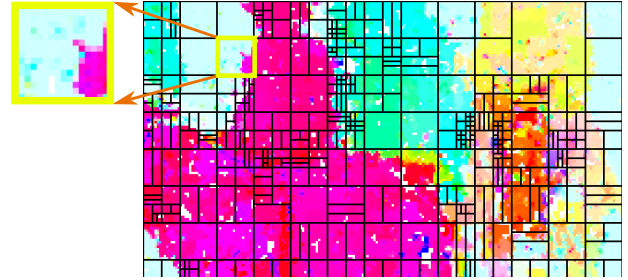
Furthermore, to reduce the problem of imbalanced data, the training datasets are composed randomly with the same amount of training instances classified into each output class.

B. Feature Evaluation and Selection

Let a feature be a property of the CU used to determine which output class shall be selected by the RF classifiers. In related works, features are extracted among others from intermediate encoding information [28], texture of pixel luminance samples [26] or motion divergence [34], as mentioned in Section II. Features based on intermediate encoding information have been shown to be effective for the S - NS classification problem in HEVC [24][28]. In this work, we choose to build a parallel friendly set of features as an input of the RF classifiers, since intense parallelization will be compulsory to achieve real time encodings for VVC standard. The features must not add dependencies between regions of the frame, in order to preserve the opportunities of high level parallelism and CU level parallelism. Therefore, the selected features are only designed based on current CU data, such as texture of pixel luminance samples and motion divergence. This choice forces to neglect features based on intermediate encoding informations. This section first introduces Motion Divergence Field (MDF), then explains the feature evaluation and selection.



(a) Original frame.



(b) Visual representation of the MDF. The MVs with different motion directions are displayed with different colors

Fig. 6. Correlation between MDF and frame QTBT partition, frame #9 *RaceHorses*, at QP=32.

1) *Motion Divergence Field*: In the following, motion divergence in a frame is considered through the MDF. The MDF is the array of MVs of every 4×4 pixels block of the frame. The MVs point to the closest reference frame in term of temporal distance. In this work, a separate motion search process is needed to compute the MDF. Without optimization and parallelization, the motion search process of the MDF induces an average 0.8% overhead of the encoding complexity. However, many real-time encoders $\times 265$ [44] already use look-ahead techniques. A look-ahead technique consists in a pre-analysis of the video sequence, generally including a motion search on small blocks of the frame. For encoders using look-ahead techniques, the overhead to compute the MDF is null.

Fig. 6a displays the original frame #9 of sequence *RaceHorses*, while Fig. 6b gives a visual representation of its MDF. MVs with different motion directions are displayed with different colors, separating visually regions of the frame with different movement. The optimal QTBT partition selected by JEM-7.0 encoder after a exhaustive RDO process is also displayed. Areas with similar colors tend to be merged together in a CU, showing the correlation between frame's optimal QTBT partition and the MDF. Some edge-examples CUs containing distant colors are recognized, and one of them is highlighted with a yellow square. Indeed, this CU is not further split and the blue zone is not separated from the purple zone. The goal of the feature selection is to determine which features extracted from the MDF and from pixel luminance samples are the most relevant to determine the best partition mode of a CU.

2) *Evaluated Features*: Evaluated features are divided into 3 categories: features computed on **whole CU**, features computed on **sub-quarters of the CU** and features based on **inconsistency among CU sub-quarters**.

Features computed on the **whole CU** are the following:

- *QP*: Quantization parameter used to encode CU slice.
- *VarPix*: Variance of luminance samples.
- *Grad*: Gradients in horizontal ($grad_x$) and vertical ($grad_y$) directions of the luminance samples (2 features).
- *RatioGrad*: ratio of gradients $\frac{grad_x}{grad_y}$.
- *VarMv*: $\sigma_{MVx}^2 + \sigma_{MVy}^2$ with σ_{MVx}^2 and σ_{MVy}^2 respectively variances of horizontal and vertical MVs of MDF.
- *MaxDiffMv*: maximum 1-norm distance between MVs of MDF, noted mv , and their mean, noted \overline{mv} , as in Equation (5).

$$\begin{aligned} \text{MaxDiffMv} &= \max_{mv \in \text{MDF}} (||mv - \overline{mv}||_1), \\ &= \max_{mv \in \text{MDF}} (|mv_x - \overline{mv}_x| + |mv_y - \overline{mv}_y|). \end{aligned} \quad (5)$$

Features based on **sub-quarters of the CU** are the following:

- *QuarterVarPix*: VarPix on 4 sub-quarters (4 features).
- *QuarterVarMv*: VarMv on 4 sub-quarters (4 features).
- *QuarterMaxDiffMv*: MaxDiffMv on 4 sub-quarters (4 features).

For any feature f , f_1 is the feature computed on top-left sub-quarter, f_2 on top-right, f_3 on bottom-left and f_4 on bottom-right. Let $\delta H(f)$ and $\delta V(f)$ be Horizontal Inconsistency (HI) and Vertical Inconsistency (VI) as defined by Equation (6)

$$\begin{aligned} \delta H(f) &= |f_1 - f_2| + |f_3 - f_4|, \\ \delta V(f) &= |f_1 - f_3| + |f_2 - f_4|. \end{aligned} \quad (6)$$

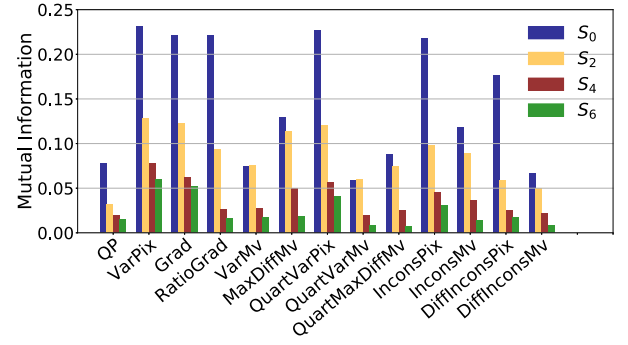
The aim of HI and VI is to highlight which rectangular parts of the CU have the highest differences. Features based on **inconsistency among sub-quarters** of the CU are the following:

- *InconsPix*: HI and VI of mean, variance and gradients-ratio of luminance samples (6 features).
- *InconsMv*: HI and VI of mean and variance of MDF (4 features).
- *DiffInconsPix*: difference between HI and VI for luminance based features (3 features).
- *DiffInconsMv*: difference between HI and VI for MDF based features (2 features).

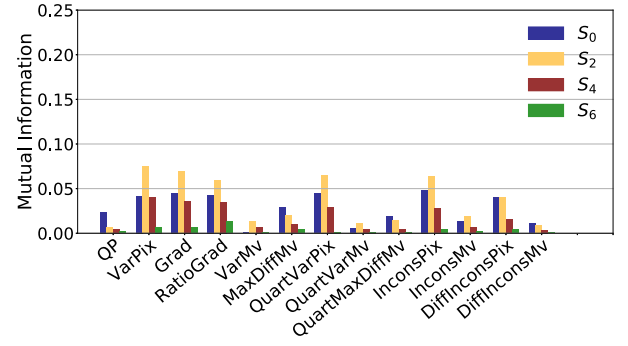
3) *Feature Selection*: As decision trees node splitting relies on MI (see Section III-A), the feature evaluation is conducted with MI as metric. Fig. 7 gives the MI of all evaluated features according to the classifier and CU size. Only MI for CU size categories S_0 , S_2 , S_4 and S_6 are displayed to avoid overloading the figure as these values are representative of MI of other CU sizes.

Fig. 7a shows that for classifier *S-NS*, the larger the CU, the higher the MI, independently of the evaluated feature. Therefore, the larger the CU, the more relevant are the evaluated features to determine whether the optimal partition mode of a CU is NoSplit or one of the Split partition modes.

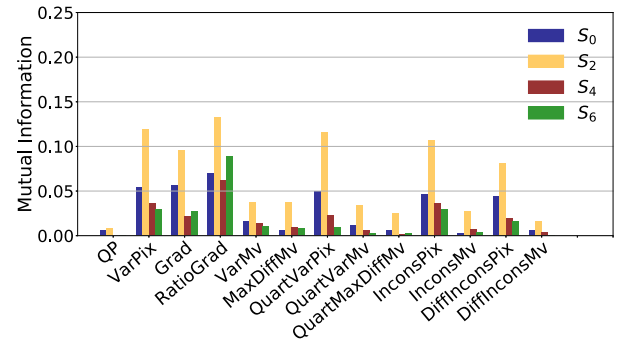
For *QT-BT* and *BH-BV* classifiers, Fig. 7b and Fig. 7c respectively show that features based on texture have higher MI than features based on the MDF, independently of CU size.



(a) *S-NS* Classifier



(b) *QT-BT* Classifier



(c) *BH-BV* Classifier

Fig. 7. Mutual Information of evaluated features according to classifier and CU sizes.

In other words, features based on texture are more relevant than features based on the MDF to estimate the partition modes to process, independently of CU size. It can also be noted that the MI of features are lower for classifiers *QT-BT* and *BH-BV* than for classifier *S-NS*. The maximum MI reaches 0.22 for classifier *S-NS* whereas the maximum MI is 0.07 and 0.14 for classifiers *QT-BT* and *BH-BV*, respectively.

For each classifier, only one set of features is selected to create the training datasets of the various CU size categories. Let the classification rate be the percentage of correct classification given by the 4-fold cross-validation on the training dataset. The selected features are those providing the highest MI and improving the classification rate when added to the set of features.

The set of features for classifier *S-NS* is composed of the 24 features: *QP*, *VarPix*, *Grad* (2 features), *RatioGrad*,

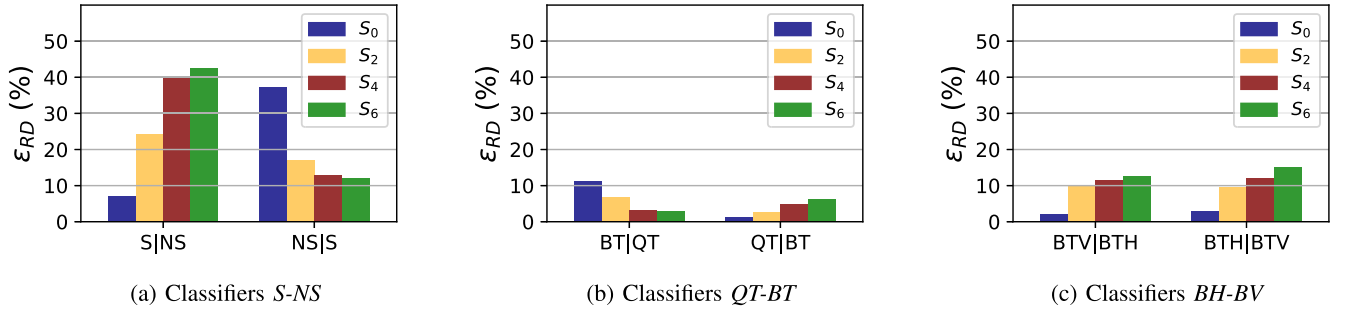


Fig. 8. Average RD misclassification error according to the classifier and CU size category. Data from 4 Sequences (*BasketballDrive*, *BQMall*, *Flowervase*, *Johnny*), encoded with $QP = 22, 27, 32, 37$.

VarMv, *MaxDiffMv*, *QuarterVarPix* (4 features), *InconsPix* (6 features), *InconsMv* (4 features), *DiffInconsPix* (3 features).

The set of features for classifier *QT-BT* is composed of the 19 features: *QP*, *VarPix*, *Grad* (2 features), *RatioGrad*, *MaxDiffMv*, *QuarterVarPix* (4 features), *InconsPix* (6 features) and *DiffInconsPix* (3 features).

For classifier *BH-BV*, the set of features is composed of the 21 features including *VarPix*, *Grad* (2 features), *RatioGrad*, *QuarterVarPix* (4 features), *QuarterVarMv* (4 features), *InconsPix* (6 features) and *DiffInconsPix* (3 features).

V. CLASSIFIERS TRAINING PROCESS

The training process consists in building the classifier through maximizing classification rate on the training dataset. In addition to classification rates, the losses of RD performance induced by misclassification are considered in the training process.

A. Impact of Misclassification on RD-Cost Errors

To assess the impact of misclassification on the encoding efficiency, the RD error ε_{RD} caused by a misclassification is introduced. In the following, the misclassification $A|B$ is when the partition mode A is chosen by the RF classifier, whereas B is the optimal partition mode selected by the encoder after exhaustive RDO process. RD error of misclassification $A|B$ is defined by Equation (7)

$$\varepsilon_{RD}(A|B) = \frac{J_A - J_B}{J_B} \quad (7)$$

J_A and J_B are the RD costs resulting from RDO process for partition modes A and B , respectively. In our case,

$$\begin{aligned} (A, B) &\in \{(NS, S), (S, NS), (QT, BT), (BT, QT), \\ &\quad (BTV, BTH), (BTH, BTV)\}, \\ J_S &= \min(J_{QT}, J_{BTH}, J_{BTV}) \text{ and} \\ J_{BT} &= \min(J_{BTH}, J_{BTV}). \end{aligned}$$

Fig. 8 shows the average ε_{RD} according to the classifier and the CU size. Results are averaged across 4 sequences (*BasketballDrive*, *BQMall*, *Flowervase*, *Johnny*) and 4 QP values.

A statistical study in Fig. 8a shows that ε_{RD} caused by classifier *S-NS* is in average 3.4 times higher compared to ε_{RD} caused by classifiers *QT-BT* and *BH-BV*. In other words,

the partition modes S and NS are in average more divergent in terms of RD cost, compared to partition modes QT and BT and partition modes BTH and BTV . Therefore *S-NS* classification problem is easier to solve compared to classification problems *QT-BT* and *BH-BV*, explaining why the MI of selected features in Section IV-B are lower for classifiers *QT-BT* and *BH-BV* compared to classifier *QT-BT*.

Concerning the first classifier *S-NS*, Fig. 8a shows that the smaller the CU, the higher $\varepsilon_{RD}(S|NS)$. When the classifier selects Split partition modes instead of the correct NoSplit partition mode selected by exhaustive RDO process, the misclassification has a higher RD impact for small CUs. On the other hand, larger CU generate higher values of $\varepsilon_{RD}(NS|S)$. This is due to the fact that the larger the CU, the greater the number of partitioning possibilities. Therefore, for large CUs, the NoSplit partition mode is more divergent in average from the optimal partitioning after exhaustive RDO process, compared to small CUs.

For the second classifier *QT-BT*, Fig. 8b shows that $\varepsilon_{RD}(BT|QT)$ is higher for large CUs (128×128 and 64×64) than small CUs (32×32 and 16×16). When the classifier selects BT partition modes instead of the correct QT partition mode, selected by exhaustive RDO process, the misclassification has stronger impact on RD cost on the large CUs than small CUs in average. Indeed, when BT partition mode is selected on large CUs, QT partition mode is no longer available, as detailed in Section II. Combined with the limit of 3 successive BT partitions, fine grain partitioning is no longer achievable. On the other hand, $\varepsilon_{RD}(QT|BT)$ is higher for small CUs than for large CUs. This is due to the fact that rectangular BT partition modes offer more partitioning shapes than square QT partition mode on small areas in the frame.

Concerning the third classifier *BH-BV*, Fig. 8c shows that ε_{RD} are symmetric for misclassification $BTH|BTV$ and $BTV|BTH$. Moreover, misclassification has very small impact in average on RD cost losses (below 2%) for S_0 category CUs and quite higher impact (around 10%) for S_2 , S_4 and S_6 category CUs.

B. Weighting of Training Dataset

Previous section shows that the impact of misclassification on RD cost losses depends highly on the classifier and the CU size. From this observation, all the training instances for classifier $A-B$ with $(A, B) \in \{(S, NS), (QT, BT), (BH, BV)\}$ are

TABLE II
CLASSIFICATION RATE (IN %) ACCORDING TO THE
CLASSIFIER AND THE CU SIZE CATEGORY

	S_0	S_1	S_2	S_3	S_4	S_5	S_6	Average
<i>S-NS</i>	83	82	78	76	73	72	69	76
<i>QT-BT</i>	69	-	70	-	67	-	60	67
<i>BH-BV</i>	62	64	70	67	69	68	67	67

assigned a weight $w(A, B)$. The value of $w(A, B)$ is computed by Eq (8).

$$\forall (A, B) \in \{(S, NS), (QT, BT), (BH, BV)\},$$

$$w(A, B) = \max(\varepsilon_{RD}(A|B), \varepsilon_{RD}(B|A)). \quad (8)$$

By assigning a weight to the training instances, the RF classifiers are built in order to minimize the sum of misclassified weights $w(A, B)$, instead of minimizing the classification error rate. Therefore the training process has more probability to classify well training instances with high weights, i.e training instances that induce high RD losses. Note that the weights are needed only during the training process, and not when the trained model is used to reduce the complexity of encoding process.

C. Classification Rates

As mentioned in Section IV-B, the classification rate is the percentage of correct classification given by the 4-fold cross-validation on the training dataset, carried out with the weighting of training instances described in Section V-B. In the following, the number of decision trees of the RFs has been set to 40 which represents a good trade-off between high classification rate and low inference time. TABLE II gives the average classification rates of the three classifiers according to the CU size category, with 40 decision trees by RF.

In TABLE II, the classification rates of classifier *S-NS* are between 69% and 83%. The larger the CU, the higher the classification rate for classifier *S-NS*. In the literature, classification rates of techniques using ML to reduce the complexity of the QT partitioning in HEVC are close to 80% [40], [45]. The classifier *S-NS* has therefore classification rate performance comparable to previous works on HEVC.

Section V-A shows that the *S-NS* classification problem is easier to solve compared to classification problems *QT-BT* and *BH-BV*. For this reason the classification rates of classifiers *QT-BT* and *BH-BV* are in average 67%, which is in 9% lower than the average classification rate of classifier *S-NS*.

VI. TUNABLE COMPLEXITY REDUCTION

In order to control RD losses induced by misclassification, risk intervals of classification are introduced for each binary classifier. In the risk interval of the binary classifier, both output partition modes are processed, limiting RD efficiency losses at the expense of complexity reduction. By varying the size of risk intervals, tunable complexity reduction is achieved.

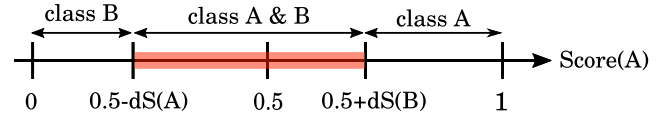


Fig. 9. Risk interval for binary classification.

A. Definition of Risk Interval

A score value, deduced from the votes of individual decision trees, is used to determine the risk interval of a given classifier. The associated score $Score(A)$ corresponds to the percentage of decision trees of the RF classifier that predicts the class *A* which is defined by Equation (9)

$$Score(A) = \frac{N_{votes}(A)}{N_{trees}} \quad (9)$$

where $N_{votes}(A)$ is the number of trees voting for class *A* and N_{trees} is the total number of trees constituting the RF classifier. $Score(A)$ takes values between 0 and 1 and the value of $Score(A)$ quantization step is $1/N_{trees}$. The closer $Score(A)$ is to 1, the more predominantly the RF classifier selects class *A*.

In our specific case, all the classification problems are between two classes *A* and *B*, with $(A, B) \in \{(NS, S), (QT, BT), (BTV, BTH)\}$. For binary classification, as $N_{votes}(A) + N_{votes}(B) = N_{trees}$, then $Score(A) + Score(B) = 1$. Using this relation, the classification decision of the binary RF classifier is *A* if $Score(A) > 0.5$ and *B* otherwise (see Section III-A).

An example of risk interval is illustrated in red color in Fig. 9. The risk interval is the range $[0.5 - dS(A), 0.5 + dS(B)]$ of $Score(A)$, with $dS(A)$ and $dS(B)$ the risk interval boundaries dS for decisions *A* and *B*, respectively. The risk interval boundary dS are included in the range $[0, 0.5]$. When $Score(A)$ is inside the risk interval, the classifier makes no decision and both output partition modes *A* and *B* are processed.

B. Computation of Risk Interval Boundaries

The values of the risk interval boundaries dS for every classifier and CU size category are computed at encoding time. Every 32 frames, 1 reference frame is encoded with the exhaustive RDO process of JEM-7.0, enabling the RF classifiers only to gather misclassification RD statistics. No complexity reduction is achieved on the encoding of the reference frames. The complexity of the reference frames is included in the final results and is compensated by the complexity reduction achieved on frames constrained by the RF classifiers.

In the reference frames, the RD costs of all partition modes are computed, making it possible to compute the sum of ε_{RD} error induced by misclassification, further called cumulative ε_{RD} . The risk interval boundaries dS are computed in order to limit cumulative ε_{RD} on the reference frame. The computation of the dS values at encoding time adjusts the RD efficiency losses to video content variations, across different sequences and scenes.

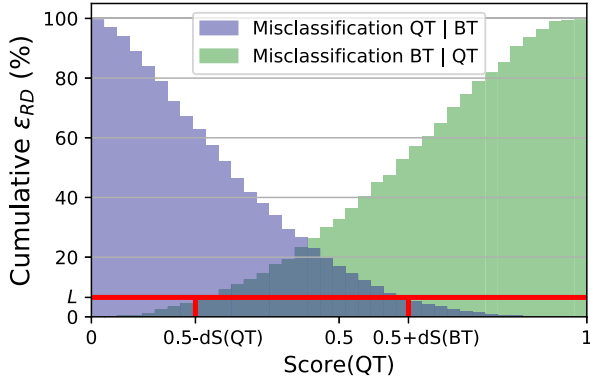


Fig. 10. Cumulative ε_{RD} of $QT|BT$ and $BT|QT$ misclassifications in function of $Score(QT)$. Data from S_6 category CUs of first frame of sequence *BasketballDrill*, $QP = 22$.

In Fig. 10, the blue and green histograms correspond to a concrete example of cumulative $\varepsilon_{RD}(QT|BT)$ and cumulative $\varepsilon_{RD}(BT|QT)$, respectively, in function of $Score(QT)$. The cumulative ε_{RD} are computed from the RD costs of S_6 category CUs, extracted from the RDO process of sequence *BasketballDrill* reference frame. The maximum cumulative ε_{RD} error tolerated on the reference frame is further noted L and represented by the red line in Fig. 10. Depending on the L value, the boundaries of the risk interval $dS(QT)$ and $dS(BT)$ are determined such as both cumulative $\varepsilon_{RD}(QT|BT)$ and cumulative $\varepsilon_{RD}(BT|QT)$ are below L . Note that $dS(QT)$ is greater than $dS(BT)$ in this example, since the errors $\varepsilon_{RD}(BT|QT)$ are greater than the errors $\varepsilon_{RD}(QT|BT)$.

C. Individual Performance of Classifiers

As explained in Section V-A, average RD losses induced by misclassification depend highly on the classifier. Reason why the possibility is left for the user to select a different threshold L by classifier, further noted L_{S-NS} , L_{QT-BT} , L_{BH-BV} . For a given classifier, the same threshold is applied on all CU size categories.

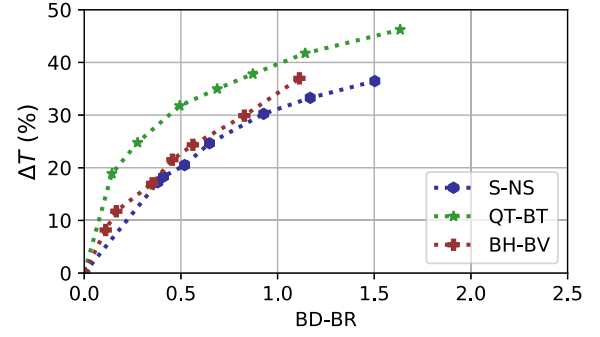
The performance of a complexity reduction solution is evaluated by measuring the trade-off between RD efficiency using the BD-BR increase [46] and encoding complexity reduction ΔT , defined by Equation (10)

$$\Delta T = \frac{1}{4} \sum_{QP_i \in \{22, 27, 32, 37\}} \frac{T_A(QP_i) - T_R(QP_i)}{T_A(QP_i)}, \quad (10)$$

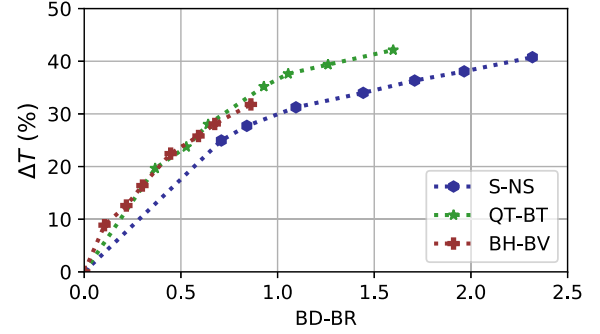
where $T_A(QP_i)$ and $T_R(QP_i)$ are the anchor (encoded with exhaustive RDO process) and reduced time required to encode the video with $QP = QP_i$, respectively.

In order to evaluate the performance of the classifiers individually, encodings are run activating only one classifier at a time with different values of L : 0.0%, 0.01%, 0.02%, 0.05%, 0.10%, 0.15%, 0.20%, 0.30%. The value $L = 0.0\%$ means that the classifier is disabled. The performance is gathered across the encodings of the 32 first frames of 10 training sequences over 4 QP values.

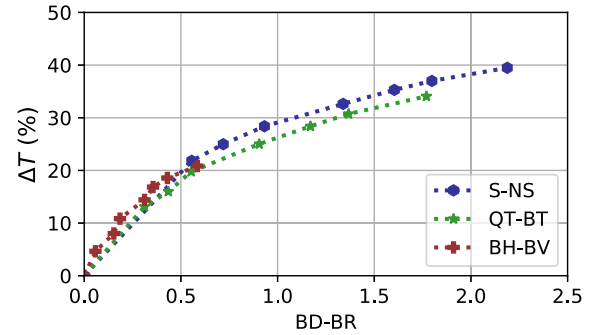
Fig. 11 shows the average ΔT versus the average BD-BR, according to the classifier for BT_{depth} equals to 1, 2 and 3.



(a) $BT_{depth} = 3$



(b) $BT_{depth} = 2$



(c) $BT_{depth} = 1$

Fig. 11. Average ΔT and BD-BR, according to the BT_{depth} and classifier. Average computed across the encodings of 32 first frames of 10 training sequences with 4 QP values. Points correspond to different values of L : 0.0%, 0.01%, 0.02%, 0.05%, 0.10%, 0.15%, 0.20%, 0.30%.

As explained in Section II-A, BT_{depth} is the encoding parameter that specifies the number of successive allowed BT partitions. In the CTC [38], BT_{depth} value is set to 3, reason why the conducted experiences only consider BT_{depth} values lower than 3. The blue, green and red curves correspond to performance obtained with the individual activation of classifiers $S-NS$, $QT-BT$ and $BH-BV$, respectively. The points of the curves are obtained from left to right for the following values of L : 0.0%, 0.01%, 0.02%, 0.05%, 0.10%, 0.15%, 0.20%, 0.30%. In Fig. 11, the higher left the curve, the better the classifier performance, as it minimizes BD-BR while maximizing ΔT .

Fig. 11a shows that for $BT_{depth} = 3$, when BD-BR is lower than 1.2%, the curve of classifier $BH-BV$ is below the curve of classifier $QT-BT$ and above the curve of classifier $S-NS$. Classifier $QT-BT$ has therefore the best performance. Even

though classifier $S\text{-}NS$ has the best successful classification rates (see Section V-C), it has the worst performance in term of trade-off between ΔT and BD-BR. This is due to the fact that misclassification induces much higher RD errors in average for classifier $S\text{-}NS$ compared to other classifiers, as explained in Section V-A.

For $BT_{depth} = 2$, Fig. 11b shows that classifiers $QT\text{-}BT$ and $BH\text{-}BV$ have equivalent performance as their curves overlap from 0% to 30% ΔT . Classifier $S\text{-}NS$ has the lowest performance as its curve is the rightmost from 0% to 40% ΔT .

Finally, when $BT_{depth} = 1$, BT partition modes are available for less CUs compared to $BT_{depth} = 3$ and $BT_{depth} = 2$. Classifiers $QT\text{-}BT$ and $BH\text{-}BV$ have therefore less CUs to address, meaning less complexity reduction opportunities compared to classifier $S\text{-}NS$. This explains why in Fig. 11c, the curve of classifier $QT\text{-}BT$ is lower compared to the curve of classifier $S\text{-}NS$ when BD-BR is between 0% and 1.7%. This also enlighten why ΔT for classifier $BH\text{-}BV$ only reaches 20%, whereas it reaches 30% for $BT_{depth} = 3$ and $BT_{depth} = 2$.

D. Optimal Selection of Complexity Reduction Configurations

In order to activate the three classifiers simultaneously at encoding time, three values of L are required. The triplet $(L_{S\text{-}NS}, L_{QT\text{-}BT}, L_{BH\text{-}BV})$ is further called Complexity Reduction Configuration (CRC).

For a given CRC, knowing the individual complexity reductions of classifiers $\Delta T(\mathbb{L}_{S\text{-}NS})$, $\Delta T(\mathbb{L}_{QT\text{-}BT})$ and $\Delta T(\mathbb{L}_{BH\text{-}BV})$ presented in Section VI-C, this section first explains how to estimate the expected complexity reduction (called ΔT_{crc}) when the three classifiers are used simultaneously. In the following, ΔT_{crc} is computed considering an example where $\Delta T(\mathbb{L}_{S\text{-}NS}) = 15\%$, $\Delta T(\mathbb{L}_{QT\text{-}BT}) = 25\%$ and $\Delta T(\mathbb{L}_{BH\text{-}BV}) = 20\%$. Intermediate values ΔT_S , ΔT_Q and ΔT_B are introduced by Equation (11)

$$\begin{aligned}\Delta T_S &= \Delta T(\mathbb{L}_{S\text{-}NS}) = 15\%, \\ \Delta T_Q &= (1.0 - \Delta T_S) \cdot \Delta T(\mathbb{L}_{QT\text{-}BT}) \\ &= 0.85 \cdot 25\% = 21\%, \\ \Delta T_B &= (1.0 - \Delta T_S - \Delta T_Q) \cdot \Delta T(\mathbb{L}_{BH\text{-}BV}) \\ &= 0.64 \cdot 20\% = 13\%.\end{aligned}\quad (11)$$

The expected complexity reduction ΔT_{crc} is given by Equation (12)

$$\Delta T_{crc} = \Delta T_S + \Delta T_Q + \Delta T_B = 49\%.\quad (12)$$

Over all CRCs achieving an expected complexity reduction ΔT_{crc} , a CRC is considered optimal when it obtains the lowest sum of BD-BR after exhaustive search. The optimal CRCs, noted from C_0 to C_4 , are given in TABLE III each corresponding to a target ΔT_{crc} , according to the BT_{depth} .

For both $BT_{depth} = 3$ and $BT_{depth} = 2$, in all optimal CRCs the value of $L_{S\text{-}NS}$ is 0.0, meaning classifier $S\text{-}NS$ is not used to reduce encoding complexity. This is explained by the results of Section VI-C, where classifier $S\text{-}NS$ has lower performance in term of trade-off between BD-BR and ΔT

TABLE III
OPTIMAL CRCs AND ASSOCIATED EXPECTED ΔT_{crc} ,
ACCORDING TO BT_{depth}

CRC Name	ΔT_{crc}	$BT_{\text{depth}} = 3$			$BT_{\text{depth}} = 2$			$BT_{\text{depth}} = 1$			
		$L_{S\text{-}NS}$	$L_{QT\text{-}BT}$	$L_{BH\text{-}BV}$	$L_{S\text{-}NS}$	$L_{QT\text{-}BT}$	$L_{BH\text{-}BV}$	$L_{S\text{-}NS}$	$L_{QT\text{-}BT}$	$L_{BH\text{-}BV}$	
C_0	30%		0.0	0.01	0.02	0.0	0.01	0.02	0.01	0.0	0.05
C_1	35%		0.0	0.05	0.02	0.0	0.02	0.05	0.01	0.0	0.15
C_2	40%		0.0	0.10	0.05	0.0	0.05	0.10	0.02	0.0	0.20
C_3	45%		0.0	0.10	0.15	0.0	0.10	0.15	0.05	0.0	0.20
C_4	50%		0.0	0.15	0.15	0.0	0.15	0.15	0.10	0.0	0.20

compared to classifiers $QT\text{-}BT$ and $BH\text{-}BV$, for $BT_{depth} = 3$ and $BT_{depth} = 2$.

When $BT_{depth} = 1$, the value of $L_{QT\text{-}BT}$ is 0.0 for all optimal CRCs. It is therefore more efficient in term of BD-BR to use only classifiers $S\text{-}NS$ and $BH\text{-}BV$ when $BT_{depth} = 1$.

VII. EXPERIMENTAL RESULTS

This section gives the experimental setup and the results obtained for the proposed tunable complexity reduction solution. Sections VII-B and VII-C present the results obtained on JEM-7.0, while Section VII-D present the results obtained on VTM-5.0.

A. Experimental Setup

The selected set test sequences is composed of 18 video sequences different from training set sequences (see Section IV-A), selecting 3 sequences by class: *Campfire*, *ParkRunning3*, *ToddlerFountain*, *PeopleOnStreet*, *SteamLocomotiveTrain*, *NebutaFestival*, *Cactus*, *RitualDance*, *Kimono*, *RaceHorsesC*, *PartyScene*, *BasketballDrill*, *ParkScene*, *KristenAndSara*, *FourPeople*, *BlowingBubbles*, *RaceHorsesD* and *BQSquare*.

The experiments are carried-out under the CTC [38] in RA coding configuration at four QP values: 22, 27, 32 and 37. The performance of the proposed complexity reduction solution is evaluated by measuring the trade-off between BD-BR increase and encoding complexity reduction ΔT , defined in Section VI-C. In the following, the complexity overhead induced by the RF inference during partition scheme is noted θ . The proposed complexity reduction solution is implemented in both JEM-7.0 and VTM-5.0. In order to limit the encoding time, JEM-7.0 encoder compares the RD cost of the whole current CU with those of the BTH and BTV partition modes to prune the QT partition mode. As our solution does not compute all the RD costs of the BT partition mode, this condition is removed in the experiments.

B. Performance Evaluation of the Proposed Solution in the JEM-7.0.

In order to set the upper bound in term of complexity reduction of the proposed solution, the maximum complexity reduction opportunity ΔT_{max} for QTBT partitioning scheme in JEM-7.0 is computed. ΔT_{max} is the value of ΔT achieved when the only tested QTBT partition is the optimal partition.

TABLE IV

BD-BR(%) AND ΔT (%) FOR JEM-7.0 EXHAUSTIVE RDO PROCESS ENCODINGS WITH DIFFERENT BT_{depth} VALUES. RESULTS AVERAGED ACROSS 18 TEST SEQUENCES

JEM-7.0 Exhaustive RDO							
$BT_{depth} = 3$		$BT_{depth} = 2$		$BT_{depth} = 1$		$BT_{depth} = 0$	
BD-BR	ΔT	BD-BR	ΔT	BD-BR	ΔT	BD-BR	ΔT
+0.0%	0%	+0.6%	20%	+2.0%	50%	+5.1%	79%

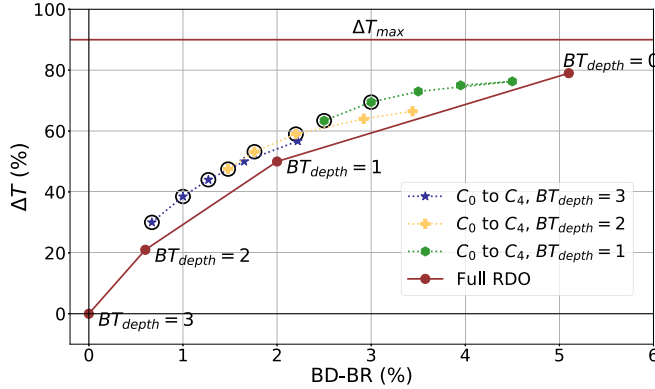


Fig. 12. Average BD-BR and ΔT for optimal CRCs and exhaustive RDO with different BT_{depth} . CRCs adopted in the proposed solution are circled in black. All results averaged across 18 test sequences and 4 QP values.

The average ΔT_{max} value across the 18 test sequences in RA configuration is:

$$\Delta T_{max} = 90\%.$$

TABLE IV shows the average BD-BR and ΔT values across the 18 test sequences, encoded with exhaustive RDO process according to the BT_{depth} value. Reducing BT_{depth} value and allowing exhaustive RDO process is a straightforward technique to reduce complexity of QTBT partition scheme. For a fixed value of BD-BR, if the average ΔT value of the proposed solution is lower than the average ΔT value obtained simply by allowing exhaustive RDO process with a reduced value of BT_{depth} , the CRC is considered as non-efficient for this value of BD-BR.

The performance of the CRCs in term of BD-BR and ΔT is displayed in Fig. 12. The upper bound ΔT_{max} , as well as BD-BR and ΔT for exhaustive RDO process with different BT_{depth} values, are also displayed in red. In Fig. 12, the blue stars correspond to CRCs with $BT_{depth} = 3$, as in the CTC. With $BT_{depth} = 3$, the optimal CRCs offer an average ΔT value between 30% and 57% for an average BD-BR increase comprised between 0.67% and 2.22%. The yellow crosses and green hexagons correspond to CRCs with reduced values $BT_{depth} = 2$ and $BT_{depth} = 1$, respectively. With $BT_{depth} = 2$, the ΔT of the optimal CRCs are in average comprised between 48% and 66% with BD-BR increase between 1.48% and 3.40%. With $BT_{depth} = 1$, the optimal CRCs offer an average ΔT value between 63% and 78% for BD-BR increase between 2.45% and 5.21%.

The 8 CRCs adopted in the proposed solution are circled in black in Fig. 12. They are located on the pareto front, i.e. the CRCs with lower BD-BR for a given value of ΔT . The CRC inducing a BD-BR increase superior

TABLE V

AVERAGE BD-BR, ΔT AND COMPLEXITY OVERHEAD θ OF THE CRCs ADOPTED IN THE PROPOSED SOLUTION

Proposed Solution in the JEM-7.0					
BT_{depth}	Adopted CRC	BD-BR (%)	ΔT (%)	θ (%)	
$BT_{depth} = 3$	C_0	0.7	30.2	1.4	
	C_1	1.0	37.3	1.2	
	C_2	1.3	44.1	1.1	
$BT_{depth} = 2$	C_0	1.6	48.2	0.7	
	C_1	1.9	54.3	0.7	
	C_2	2.3	59.4	0.6	
$BT_{depth} = 1$	C_0	2.5	63.2	0.5	
	C_1	3.0	70.0	0.5	

than 3% are not adopted in the proposed solution since they offer a trade-off between ΔT and BD-BR not considered good enough. The adopted CRCs include: $C_0(BT_3)$, $C_1(BT_3)$, $C_2(BT_3)$, $C_0(BT_2)$, $C_1(BT_2)$, $C_2(BT_2)$, $C_0(BT_1)$ and $C_1(BT_1)$. TABLE V summarizes the average BD-BR, ΔT and θ of the CRCs adopted in the proposed solution. TABLE V shows that in order to achieve complexity reductions higher than 43% in average, it is more efficient to apply our solution with a value of $BT_{depth} < 3$, compared to applying our solution with $BT_{depth} = 3$. With these adopted CRCs, the proposed tunable solution offers a range of average ΔT between 30% and 70% for an average BD-BR increase between 0.7% and 3.0%.

C. Comparison With Related Works in JEM-7.0.

The proposed solution is evaluated and compared to previous techniques on QTBT partition scheme in RA configuration [29], [31], [33] and [34]. Previous techniques [29] and [31] offer an average encoding complexity reduction of 17% and 10% for an average BD-BR increase of 0.5% and 0.2%, respectively. The encoding complexity reductions are much lower compared to the encoding complexity reductions proposed in our solution, which is in minimum equals to 30% for $C_0(BT_3)$.

For a fairer comparison with the two most recent techniques [33] and [34], TABLE VI details the performance of 2 CRCs of the proposed solution - $C_0(BT_3)$ and $C_0(BT_1)$ - with respect to the performance announced by Wang et al. in papers [33] and [34]. The performance in TABLE VI is shown by test sequence, in terms of BD-BR increase, encoding complexity reduction ΔT and θ induced by the respective techniques.

TABLE VI shows that configuration $C_0(BT_3)$ applied to the same sequences as technique [33], offers in average the same encoding complexity reduction ($\Delta T \approx 33\%$) for a BD-BR increase 0.12% higher in average. However, technique [33] is based on CNNs to reduce the encoding complexity without specifying his implementation, whereas CNNs are known to have high computational overhead. For $C_0(BT_3)$, θ has values between 0.7% and 1.8%, whereas θ for technique [33] has values between 1.7% and 4.2% according to the sequence. θ is included in the encoding complexity reductions of of the proposed solution. These overhead performance confirms the lightweight of our approach and highlights that RF classifiers

TABLE VI

BD-BR, ΔT AND COMPLEXITY OVERHEAD θ OF CRCs $C_0(BT_3)$ AND $C_0(BT_1)$ OF THE PROPOSED SOLUTION WITH RESPECT TO THE PERFORMANCE ANNOUNCED BY WANG *et al.* IN PAPERS [33] AND [34]. THE RESULTS ARE SHOWN BY TEST SEQUENCE

Class	Sequence name	Proposed Solution CRC $C_0(BT_3)$			Related Work Wang CNN [33]			Proposed Solution CRC $C_0(BT_1)$			Related Work Wang Proba [34]	
		BD-BR (in %)	ΔT (in %)	θ (in %)	BD-BR (in %)	ΔT (in %)	θ (in %)	BD-BR (in %)	ΔT (in %)	θ (in %)	BD-BR (in %)	ΔT (in %)
A1	<i>Campfire</i>	0.36	35.1	0.5	0.66	40.6	2.4	3.4	68.8	0.2	1.7	46.7
A1	<i>ParkRunning3</i>	0.49	21.1	1.0	-	-	-	2.0	57.4	0.4	-	-
A1	<i>ToddlerFountain</i>	0.58	18.2	0.8	-	-	-	1.8	59.5	0.3	1.2	48.4
A2	<i>PeopleOnStreet</i>	0.22	21.8	1.5	-	-	-	2.2	58.0	0.5	-	-
A2	<i>SteamLocomotiveTrain</i>	0.57	24.6	1.0	-	-	-	2.7	63.5	0.3	-	-
A2	<i>NebutaFestival</i>	1.37	35.1	0.9	-	-	-	2.2	68.0	0.3	-	-
B	<i>Cactus</i>	1.18	34.1	1.2	-	-	-	2.8	64.1	0.4	1.5	48.2
B	<i>RitualDance</i>	0.90	32.0	0.7	0.55	32.6	4.2	3.9	64.7	0.2	-	-
B	<i>Kimono</i>	0.79	23.1	1.0	-	-	-	4.0	65.3	0.3	1.4	54.7
B	<i>ParkScene</i>	0.63	30.8	1.6	-	-	-	2.5	68.1	0.5	1.2	45.7
C	<i>RaceHorsesC</i>	0.42	32.8	0.9	0.47	30.7	1.7	2.1	63.1	0.3	1.6	60.5
C	<i>PartyScene</i>	0.45	29.2	1.5	0.54	34.6	2.2	1.6	62.8	0.5	1.7	62.3
C	<i>BasketballDrill</i>	0.61	30.2	1.2	-	-	-	2.4	63.5	0.4	1.5	62.3
D	<i>RaceHorsesD</i>	0.41	28.8	1.2	0.51	36.3	2.7	2.3	61.5	0.6	1.3	56.5
D	<i>BQSquare</i>	0.64	25.1	1.8	0.44	26.0	2.1	1.5	57.5	0.8	1.3	51.8
D	<i>BlowingBubbles</i>	0.46	31.6	1.6	0.60	35.7	1.9	1.9	60.0	0.8	1.2	50.3
E	<i>FourPeople</i>	0.76	40.8	1.8	0.32	28.9	4.1	2.5	68.3	0.5	1.3	47.7
E	<i>KristenAndSara</i>	1.17	46.6	1.2	0.38	33.8	3.6	2.7	70.2	0.4	1.0	45.6
Same Sequences Average		0.62	33.5	1.4	0.50	32.8	2.8	2.42	64.0	0.5	1.38	52.3
Global Average		0.67	30.2	1.3				2.45	63.4	0.5		

consume few computing resources, which is a key point to use this solution in a real-time or embedded framework.

TABLE VI also shows that configuration $C_0(BT_1)$ achieves higher encoding complexity reductions for all tested sequences compared to considered previous techniques, and achieves in average 12% higher encoding complexity reduction compared to technique [34], with tolerable BD-BR increase of 2.45% in average. Moreover, the fact that the proposed solution is tunable offers more flexibility for concrete use-cases compared to previous techniques that aim to reduce the complexity of QTBT partitioning scheme.

D. Performance Evaluation of the Proposed Solution in the VTM-5.0

The VTM-5.0 is the latest reference software for VVC standardization. Several new coding tools have been added compared to the JEM-7.0 reference software. For instance, the VTM-5.0 includes the MTT partitioning scheme, more complex than QTBT in JEM-7.0. Thus, the proposed solution has also been implemented in the VTM-5.0 in order to verify its performance. The following section first introduces the MTT partitioning scheme and second presents the results of the proposed solution integrated in the VTM-5.0.

The MTT partitioning scheme is an extension to QTBT that enables Ternary Tree (TT) partition modes, including Ternary Tree Horizontal (TTH) partition mode and Ternary Tree Vertical (TTV) partition mode. When TT partition modes are used, the CU is divided either horizontally or vertically into three blocks and the size of the middle block is half the size of the CU, as shown in Fig. 13. The MT_{depth} parameter defines

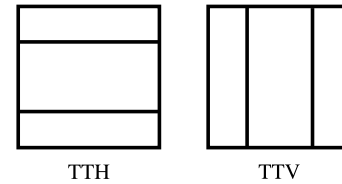


Fig. 13. Additional partition modes in MTT partitioning scheme.

the maximum number of successive BT or TT partitions allowed for the encoding of a CTU.

The proposed solution has originally been designed for complexity reduction of QTBT partitioning scheme. MTT partitioning scheme is more complex than QTBT partitioning scheme, as it enables two additional partition modes. In order to adapt the proposed solution to MTT partitioning scheme, horizontal partition modes including TTH and BTH, and vertical partition modes including TTV and BTV, are both grouped as outputs of the $BH-BV$ classifier, as shown in Figure 14. The same classifier $BH-BV$ is used to classify both BT and TT partition modes in the VTM-5.0. This choice is supported by the fact that partition modes TTH and BTH, as well as partition modes TTV and BTV, generate partitions with the same directions.

The same features selected in Section IV-B are used as an input of the RF classifiers. Moreover, CRC assessed under the VTM-5.0 are selected with the same process described in Section VI-D. Instead of applying the CRCs with different values of BT_{depth} as in JEM-7.0, the CRCs are applied with different values of MT_{depth} in the VTM-5.0. The performance of the CRCs in term of BD-BR and ΔT is illustrated in

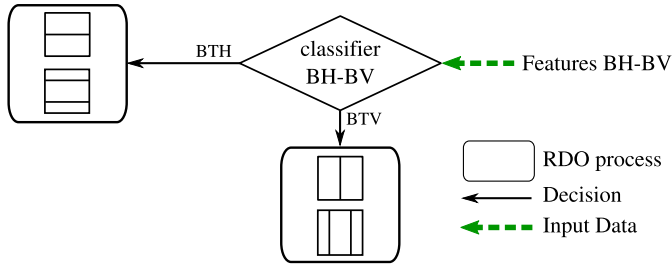


Fig. 14. Outputs modification of BH-BV classifier in VTM-5.0.

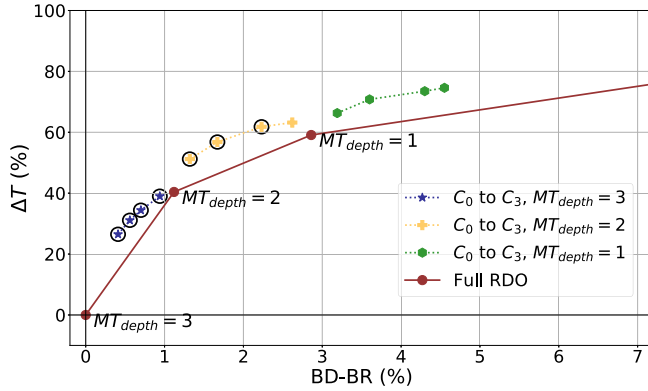
Fig. 15. Average BD-BR and ΔT for optimal CRCs and exhaustive RDO with different MT_{depth} . CRCs adopted in proposed solution are circled in black. All results averaged across 18 test sequences and 4 QP values.

Fig. 15. The BD-BR and ΔT for exhaustive RDO process with different MT_{depth} values, are also displayed in red. In Fig. 15, the blue stars correspond to CRCs with $MT_{depth} = 3$, while the yellow crosses and green hexagons correspond to CRCs with reduced values $MT_{depth} = 2$ and $MT_{depth} = 1$, respectively.

Fig. 15 shows that for a similar ΔT complexity reduction, the proposed solution offers a BD-BR value 1.0% lower compared to exhaustive RDO process with $MT_{depth} = 1$. This result confirms that our approach is relevant compared to the most straightforward complexity reduction technique, that allows exhaustive RDO process with reduced values of MT_{depth} .

The 7 CRCs circled in black in Fig. 12 are adopted in the proposed solution. As for the JEM-7.0 in Section VII-B, the CRC inducing a BD-BR increase superior than 3% are not adopted in the proposed solution since they do not offer a relevant trade-off between ΔT and BD-BR. TABLE VII summarizes the average BD-BR, ΔT and θ of the CRCs adopted in the proposed solution. TABLE VII shows that the complexity reductions vary from 25% to 61% in average for 0.4% to 2.2% BD-BR increase. Moreover, the RF inference overhead θ is below 0.7% for all the CRCs, which confirms the lightweight overhead of our approach.

TABLE VIII shows the BD-BR, ΔT and θ of CRCs $C_1(MT_3)$ and $C_0(MT_2)$ of the proposed solution under the VTM-5.0, according to the test sequence. We can notice that the scores given in TABLE VIII differ slightly according to the test sequence. For instance, the lowest resolution sequences (class D), including *RaceHorsesD*, *BQSquare* and *BlowingBubbles*, achieves in average 54.6% ΔT for

TABLE VII
AVERAGE BD-BR, ΔT AND COMPLEXITY OVERHEAD θ OF THE CRCs ADOPTED IN THE PROPOSED SOLUTION FOR VTM-5.0

Proposed Solution in the VTM-5.0				
Adopted CRC		BD-BR (%)	ΔT (%)	θ (%)
$MT_{depth} = 3$	C_0	0.43	25.5	0.6
	C_1	0.61	30.1	0.6
	C_2	0.75	33.4	0.6
	C_3	0.97	38.6	0.5
$MT_{depth} = 2$	C_0	1.32	50.7	0.3
	C_1	1.67	56.3	0.3
	C_2	2.22	61.5	0.3

TABLE VIII
BD-BR, ΔT AND COMPLEXITY OVERHEAD θ OF CRCs $C_1(MT_3)$ AND $C_0(MT_2)$ OF THE PROPOSED SOLUTION IN THE VTM-5.0, ACCORDING TO THE TEST SEQUENCE

Sequence name	Proposed Solution CRC $C_1(MT_3)$			Proposed Solution CRC $C_0(MT_2)$		
	BD-BR (in %)	ΔT (in %)	θ (in %)	BD-BR (in %)	ΔT (in %)	θ (in %)
<i>Campfire</i>	0.95	29.3	0.4	3.06	65.1	0.2
<i>ParkRunning3</i>	0.46	30.9	0.4	1.91	62.9	0.2
<i>ToddlerFountain</i>	0.74	23.2	0.4	1.95	64.2	0.2
<i>PeopleOnStreet</i>	0.72	25.6	0.6	3.13	62.2	0.2
<i>SteamLocomotive</i>	0.92	44.9	0.4	1.71	69.5	0.2
<i>NebutaFestival</i>	0.25	36.3	0.5	0.83	75.8	0.2
<i>Cactus</i>	1.00	36.7	0.5	2.45	64.5	0.2
<i>RitualDance</i>	0.93	27.5	0.5	3.26	61.5	0.2
<i>Kimono</i>	1.16	33.7	0.5	2.74	64.7	0.2
<i>ParkScene</i>	0.29	30.3	0.6	1.93	59.2	0.3
<i>RaceHorsesC</i>	0.35	26.9	0.6	2.77	64.1	0.2
<i>PartyScene</i>	0.42	19.6	0.8	2.05	58.0	0.3
<i>BasketballDrill</i>	1.00	29.9	0.7	2.96	62.8	0.3
<i>RaceHorsesD</i>	0.17	26.6	0.7	2.19	57.1	0.4
<i>BQSquare</i>	0.16	26.2	1.0	1.55	49.1	0.6
<i>BlowingBubbles</i>	0.54	31.3	0.8	1.79	57.7	0.5
<i>FourPeople</i>	0.46	31.6	0.7	1.90	55.6	0.5
<i>KristenAndSara</i>	0.53	30.5	0.6	1.74	54.3	0.4
Average	0.61	30.1	0.6	2.22	61.5	0.3

$C_2(MT_2)$ configuration. The ΔT values of class D sequences are in average 7% lower compared to the average ΔT value of all test sequences for $C_2(MT_2)$. Indeed, the lowest resolution sequences tend to have a finer grained partitioning, which offers less complexity reduction opportunities compared to higher resolution sequences.

In conclusion, this section has shown that it is possible to apply the proposed solution in the VTM-5.0. The performance under the VTM-5.0 prove that the proposed solution is scalable to two different encoders and does not over-fit the JEM-7.0 encoding characteristics. The good performance achieved under VTM-5.0, the latest reference software of VVC standard, also attests the reliability of the proposed solution for future encoders, compliant with VVC standard. Finally, to the best of our knowledge, this contribution is the first to propose a complexity reduction technique for the VTM reference software in Inter coding configuration.

VIII. CONCLUSION

In this paper, a tunable ML solution based on RF classifiers to speed up the QTBT partitioning scheme in RA configuration

is proposed. Three binary RF classifiers are trained off-line in order to ignore expensive exploration of the partition modes classified as unlikely. By varying the size of risk intervals for classification decision, tunable complexity reduction is achieved, offering an average encoding complexity reduction varying from 30% and 70% for an average BD-BR increase between 0.7% and 3.0% in the JEM-7.0, with very low overhead. The proposed solution as also been implemented in the JVET software post JEM, named VTM. To this end, the proposed solution as been extended to the new TT partition modes included in the VTM partition scheme. In VTM-5.0 software, encoding complexity reductions vary from 25% to 61% in average for only 0.4% to 2.2% BD-BR increase.

Tunable encoding complexity reduction being the first step for encoding time control, future works will investigate the possible modification in the proposed solution in order to achieve encoding time control.

REFERENCES

- [1] CISCO. (2016). *Global_2021_forecast_highlights*. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf
- [2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [4] 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.
- [5] T. K. Tan *et al.*, "Video quality evaluation methodology and verification testing of HEVC compression performance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 76–90, Jan. 2016.
- [6] N17055, *Algorithm Description of Joint Exploration Test Model 7 (JEM 7)*. Accessed: Jul. 2017. [Online]. Available: <https://mpeg.chiariglione.org/standards/exploration/future-video-coding/n17055-algorithm-description-joint-exploration-test-model>
- [7] *Joint Exploration Model (JEM) Reference Software 7.0*. Accessed: Aug. 2016. [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/branches/HM-16.6-JEM-7%0-dev/
- [8] N. Sidaty, W. Hamidouche, O. Deforges, and P. Philippe, "Emerging video coding performance: 4K quality monitoring," in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Erfurt, Germany, May 2017, pp. 1–3.
- [9] J. An, H. Han, and Z. Kai, *Quadtree Plus Binary Tree Structure Integration with JEM Tools* document JVET-B0023, Joint Video Exploration Team (JVET), Feb. 2016.
- [10] W. Jiang, H. Ma, and Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," in *Proc. 2nd Int. Conf. Consum. Electron., Commun. Netw.*, Xianning, China, Apr. 2012, pp. 1836–1840.
- [11] T. L. da Silva, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC intra prediction mode decision based on EDGE direction information," in *Proc.*, Aug. 2012, pp. 1–5.
- [12] T. Zhang, M.-T. Sun, D. Zhao, and W. Gao, "Fast intra-mode and CU size decision for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1714–1726, Aug. 2017.
- [13] S. Ryu and J. Kang, "Machine learning-based fast angular prediction mode decision technique in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5525–5538, Nov. 2018.
- [14] Z. Pan, S. Kwong, M.-T. Sun, and J. Lei, "Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 405–412, Jun. 2014.
- [15] P.-T. Chiang and T. S. Chang, "Fast zero block detection and early CU termination for HEVC video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 1640–1643.
- [16] G. Correa, P. Assuncao, L. Agostini, and L. D. S. Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.
- [17] G. Corrêa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Coding tree depth estimation for complexity reduction of HEVC," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2013, pp. 43–52.
- [18] H. Lee, H. J. Shim, Y. Park, and B. Jeon, "Early skip mode decision for HEVC encoder with emphasis on coding quality," *IEEE Trans. Broadcast.*, vol. 61, no. 3, pp. 388–397, Sep. 2015.
- [19] B. Min and R. C. C. Cheung, "A fast CU size decision algorithm for the HEVC intra encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 892–896, May 2015.
- [20] L. Shen, Z. Zhang, and Z. Liu, "Effective CU size decision for HEVC intracoding," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232–4241, Oct. 2014.
- [21] A. Mercat, F. Arrestier, M. Pelcat, W. Hamidouche, and D. Menard, "Prediction of quad-tree partitioning for budgeted energy HEVC encoding," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2017, pp. 1–6.
- [22] L. Shen, Z. Liu, Z. Zhang, and X. Shi, "Fast inter mode decision using spatial property of motion field," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1208–1214, Oct. 2008.
- [23] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.
- [24] S. G. Blasi, I. Zupancic, E. Izquierdo, and E. Peixoto, "Fast HEVC coding using reverse CU visiting," in *Proc. Picture Coding Symp. (PCS)*, May 2015, pp. 50–54.
- [25] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, "CU partition mode decision for HEVC hardwired intra encoder using convolution neural network," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5088–5103, Nov. 2016.
- [26] F. Duanmu, Z. Ma, and Y. Wang, "Fast CU partition decision using machine learning for screen content compression," in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, QC, Canada, Sep. 2015, pp. 4972–4976.
- [27] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP J. Image Video Process.*, vol. 4, no. 1, pp. 1–11, Dec. 2013.
- [28] X. Shen, L. Yu, and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," in *Proc. Picture Coding Symp.*, May 2012, pp. 453–456, IEEE.
- [29] Y. Yamamoto, *AHG5: Fast QTBT Encoding Configuration*, document JVET-D0095, Oct. 2016.
- [30] H. Huang, S. Liu, and Y.-W. Huang, *Ahg5: Speed-Up for JEM-3.1*, document JVET-D0077, Oct. 2016.
- [31] P.-H. Lin, C.-L. Lin, and Y.-J. Chang, *AHG5: Enhanced Fast Algorithm of JVET-E0078*, document JVET-F0063, Apr. 2017.
- [32] Z. Jin, P. An, L. Shen, and C. Yang, "CNN oriented fast QTBT partition algorithm for JVET intra coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4, IEEE.
- [33] Z. Wang, S. Wang, X. Zhang, S. Wang, and S. Ma, "Fast QTBT partitioning decision for interframe coding with convolution neural network," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2550–2554.
- [34] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Probabilistic decision based block partitioning for future video coding," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1475–1486, Mar. 2018.
- [35] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [36] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proc. CVPR Jun.* 2011, pp. 617–624.
- [37] J. An, Y.-W. Chen, K. Zhang, H. Huang, Y.-W. Huang, and S. Lei. (2015). *Block Partitioning Structure for Next Generation Video Coding*. Accessed: Sep. 2015. [Online]. Available: <https://www.itu.int/md/T13-SG16-C-0966/en>
- [38] J. Boyce, K. Suehring, and X. Li, "JVET-J1010: JVET common test conditions and software reference configurations," JVET, San Diego, CA, USA, Tech. Rep. JVET-J1010, 2018.
- [39] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin, "Joint separable and non-separable transforms for next-generation video coding," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2514–2525, May 2018.
- [40] A. Mercat *et al.*, "Machine learning based choice of characteristics for the one-shot determination of the HEVC intra coding tree," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 263–267.
- [41] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Apr. 2001.

- [42] C. B. Ross, "Mutual information between discrete and continuous data sets," *PLoS One*, vol. 9, no. 2, Feb. 2014, Art. no. e87357.
- [43] *Subjective Video Quality Assessment Methods for Multimedia Applications*, document ITU-T P.910, 1999.
- [44] x265. [Online]. Available: <https://bitbucket.org/multicoreware/x265>
- [45] G. Correa, A. P. Assuncao, L. V. Agostini, and A. L. da Silva Cruz, "Fast HEVC encoding decisions using data mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 660–673, Apr. 2015.
- [46] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33 ITU-T, 2001.



Thomas Amestoy received the M.Sc. degree in mathematical modeling, image and simulation from the ENSIMAG Engineering School, Grenoble, France, in 2015. He is currently pursuing the industry-oriented Ph.D. degree in partnership between a French Company, HTE/STR/MMP Gennevilliers, Thales SIX GTS France, France, and also with the INSA Rennes, Institute of Electronics and Telecommunications of Rennes (IETR), UMR CNRS 6164.

From 2015 to 2017, he was an Engineer with Scalian, Toulouse and Rennes, France.



Alexandre Mercat received the Ph.D. degree in electrical and computer engineering from the Institute of Electronics and Telecommunications of Rennes (IETR) Laboratory, Rennes, France, in 2018. His Ph.D. dissertation focuses on the complexity reduction and control of the HEVC encoding process. He has been a Postdoctoral Researcher with the Computing Sciences, Tampere University, Tampere, Finland, since 2018. His research interests include implementation of image and signal processing applications in many core embedded systems,

real-time implementations of the new generation video coding standards, complexity-aware video coding, machine learning, approximate computing, power consumption, and digital systems design.



Wassim Hamidouche received the Ph.D. degree in signal and image processing from the University of Poitiers, France, in 2010. From 2011 to 2012, he was a Research Engineer with the Canon Research Centre, Rennes, France. He has been an Associate Professor with INSA Rennes, since 2015, and a member of the Institute of Electronics and Telecommunications of Rennes (IETR), UMR CNRS 6164. His research interests include video coding, real-time implementations of the new generation video coding standards, multimedia quality assessment, and multimedia content security.



Daniel Menard received the Ph.D. and HDR (habilitation to conduct researches) degrees in signal processing and telecommunications from the University of Rennes in 2002 and 2011, respectively. From 2003 to 2012, he was an Associate Professor with the ECE Department, University of Rennes Engineering School, ENSSAT. He was also a member of the IRISA/INRIA Laboratory. He is currently a Professor with the Electronics and Computer Engineering (ECE) Department, Graduate Engineering School, INSA Rennes. He is also a member of the IETR/CNRS Laboratory. He has authored over 80 international papers distributed in the areas of embedded systems, computer-aided design, and arithmetic and signal processing. His research interests include implementation of image and signal processing applications in embedded systems, approximate computing, fixed-point arithmetic, low power systems, and video compression.



Cyril Bergeron received the degree in electrical engineering from the Ecole Supérieure d'Ingenieurs de Nice, Sophia-Antipolis, Nice, France, in 2001, the Diplôme d'Etudes Approfondies degree in signal and image processing from the Ecole Doctorale de Nice, Sophia-Antipolis, in 2003, and the Ph.D. degree from the Image and Signal Processing Department, Telecom ParisTech, Paris, France, in 2007. He joined Thales SIX GTS France, Gennevilliers, France, in 2003. His current research interests include video compression, advanced video coding, high efficiency video coding, selective encryption, unequal error protection, and joint source and channel decoding techniques.